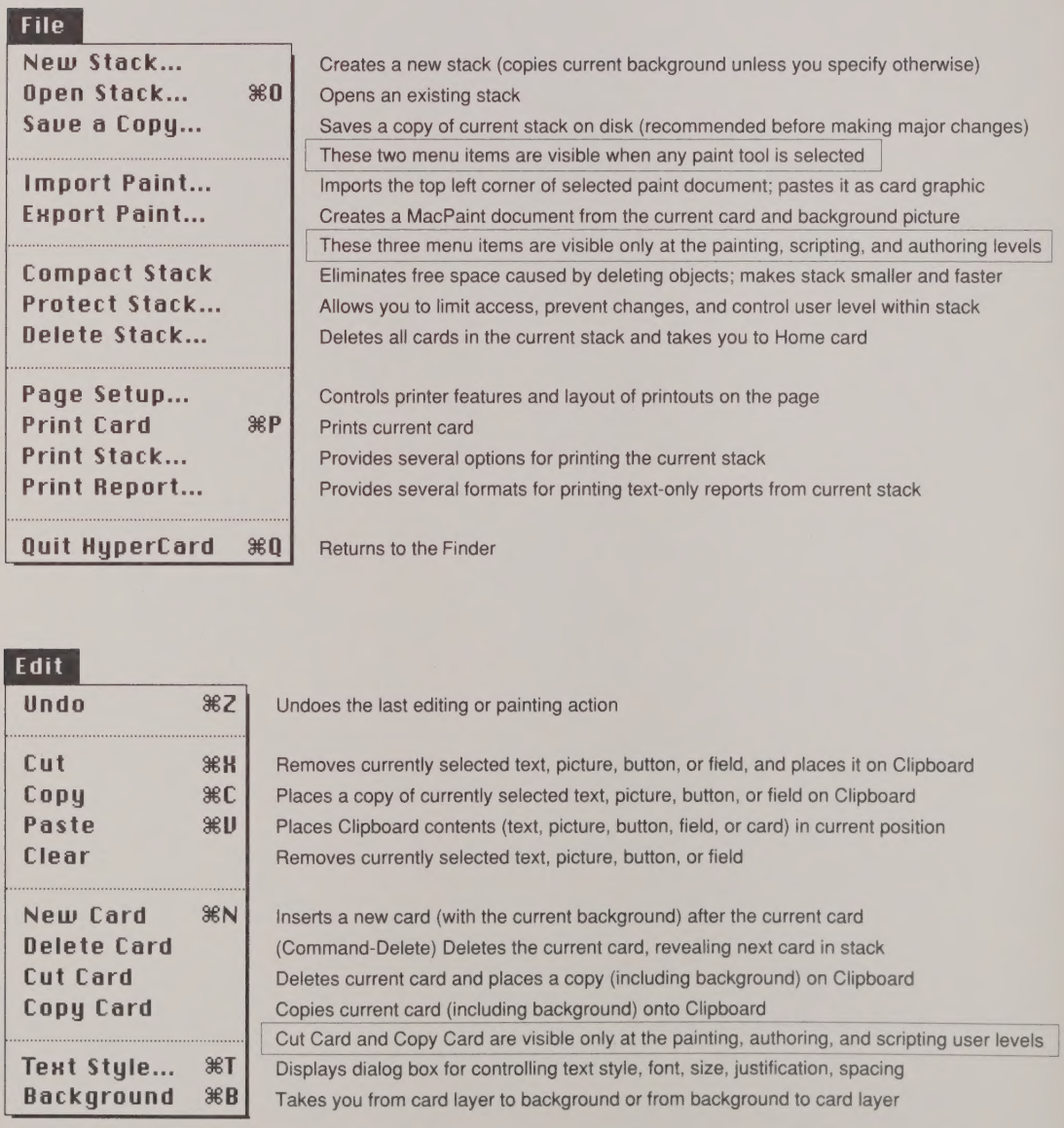
HYPERCARD IN A HURRY

GEORGE BEEKMAN

Foreword by Danny Goodman


File
New Stack...
Open Stack... ⌘O
Save a Copy...
Import Paint...
Export Paint...
Compact Stack
Protect Stack...
Delete Stack...
Page Setup...
Print Card ⌘P
Print Stack...
Print Report...
Quit HyperCard ⌘Q
Creates a new stack (copies current background unless you specify otherwise)
Opens an existing stack
Saves a copy of current stack on disk (recommended before making major changes)
These two menu items are visible when any paint tool is selected
Imports the top left corner of selected paint document; pastes it as card graphic
Creates a MacPaint document from the current card and background picture
These three menu items are visible only at the painting, scripting, and authoring levels
Eliminates free space caused by deleting objects; makes stack smaller and faster
Allows you to limit access, prevent changes, and control user level within stack
Deletes all cards in the current stack and takes you to Home card
Controls printer features and layout of printouts on the page
Prints current card
Provides several options for printing the current stack
Provides several formats for printing text-only reports from current stack
Returns to the Finder
Edit
Undo ⌘Z
Cut ⌘X
Copy ⌘C
Paste ⌘V
Clear
New Card ⌘N
Delete Card
Cut Card
Copy Card
Text Style... ⌘T
Background ⌘B
Undoes the last editing or painting action
Removes currently selected text, picture, button, or field, and places it on Clipboard
Places a copy of currently selected text, picture, button, or field on Clipboard
Places Clipboard contents (text, picture, button, field, or card) in current position
Removes currently selected text, picture, button, or field
Inserts a new card (with the current background) after the current card
(Command-Delete) Deletes the current card, revealing next card in stack
Deletes current card and places a copy (including background) on Clipboard
Copies current card (including background) onto Clipboard
Cut Card and Copy Card are visible only at the painting, authoring, and scripting user levels
Displays dialog box for controlling text style, font, size, justification, spacing
Takes you from card layer to background or from background to card layer

Go

Back	⌘~	Takes you back one card (to the one you most recently visited)
Home	⌘H	Takes you to the first card of the Home stack
Help	⌘?	Takes you to the first card of the HyperCard Help stack
Recent	⌘R	Presents the 42 most recent cards in a dialog box; click on one to go there
First	⌘1	Takes you to the first card of the current stack
Prev	⌘2	Takes you to the card before this card in the current stack
Next	⌘3	Takes you to the next card in the current stack
Last	⌘4	Takes you to the last card in the current stack
Find...	⌘F	Displays the message box containing Find command; type characters and press Return
Message	⌘M	Displays the message box; type command or message and press Return

Objects

Visible only at authoring and scripting user levels when Browse, Button, or Field tool is selected

Button Info...	(Double-click on button with Button tool selected) Displays Button Info dialog box
Field Info...	(Double-click on field with Field tool selected) Displays Field Info dialog box
Card Info...	Displays Card Info dialog box
Bkgnd Info...	Displays Background Info dialog box
Stack Info...	Displays Stack Info dialog box

If you hold down Shift while selecting any of these five commands, you're taken directly to the script

Bring Closer	⌘+	Brings a selected button or field one level closer
Send Farther	⌘-	Sends the selected button or field one level farther
New Button		Creates a new button called New Button; double-click it to display dialog box
New Field		Creates a new field; double-click it to display dialog box
New Background		Creates a new card with a new, blank background

Power Keys↓

Options

Visible only when a paint tool is selected

G	Grid	Constrains paint tools so they work along the lines of an invisible grid (toggles on/off)
	FatBits	Zooms in for detailed graphic work; Option key turns pointer to grabber hand
	Power Keys	Allows single keystrokes to activate paint tools and commands (shown to left of menus here)
1-4, 6, 8	Line Size...	Displays dialog box for selecting line size for lines, polygons, ovals, rectangles
	Brush Shape...	Displays dialog box for selecting paintbrush shape
	Edit Pattern...	Displays dialog box for editing selected pattern
	Polygon Sides...	Displays dialog box for selecting the number of sides of regular polygon
	Draw Filled	Causes shape tools to draw shapes filled with selected pattern (toggles on/off)
C	Draw Centered	Causes shape tools to draw shapes from center rather than edge (toggles on/off)
M	Draw Multiple	Causes shape tools to draw repeated images automatically (toggles on/off)

Patterns

Visible only when a paint tool is selected

W

B

Other patterns have no power key equivalents

HyperCard in a Hurry

George Beekman
Oregon State University

Wadsworth Publishing Company
Belmont, California
A Division of Wadsworth, Inc.

Computer Science Editor: Frank Ruggirello
Editorial Assistant: Carol Carreon
Production Editor: Leland Moss
Managing Designer: James Chadwick
Print Buyer: Barbara Britton
Compositor and Design: Publication Services
Copy Editor: Anne Draus
Signing Representative: Tamy Stenquist

© 1990 by Wadsworth, Inc. All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Wadsworth Publishing Company, Belmont, California 94002, a division of Wadsworth, Inc.
Apple, the Apple logo, ImageWriter, LaserWriter, and MacPaint are registered trademarks of Apple Computer, Inc. HyperCard, HyperTalk, Macintosh, MultiFinder, and Stackware are trademarks of Apple Computer, Inc.
Glossary reproduced with the permission of Apple Computer, Inc.
© 1987, Apple Computer, Inc.

Printed in the United States of America 49

1 2 3 4 5 6 7 8 9 10—94 93 92 91 90

Library of Congress Cataloging-in-Publication Data
Beekman, George.
HyperCard in a hurry / George Beekman.
p. cm.
ISBN 0-534-13038-0
1. Macintosh (Computer) —Programming. 2. Hypercard (Computer program) I. Title.
QA76.8.M 3B43 1990
005.265—dc20 89-70609
CIP

CONTENTS

FOREWORD ix

PREFACE xiii

ACKNOWLEDGMENTS xvi

INTRODUCTION xvii

What Is HyperCard? xvii
HyperCard, Hypertext, and Hypermedia: Beyond Books xix
HyperCard Essentials xx
Charting a Path through This Book xx

SESSION 1 HYPERCARD FOR BROWSERS:
A CARD FOR EVERY TASTE 2

Objectives 2
The Problem 3
Introduction 4
Customizing This Session 4
Meet Mac 7
Hello, HyperCard 14
HyperCard Navigation Tips 26

Summary 32
Key Words 32
Self-Testing Exercises 33

SESSION 2 THE DYNAMIC FILE CABINET: INFORMATION STORAGE AND RETRIEVAL 34

Objectives 34
The Problem 35
Introduction 36
Stacks, Macs, and Backups 37
Home Base 45
HyperCard's Text Fields 50
Basic Database Operations with HyperCard 57
Summary 64
Key Words 64
Self-Testing Exercises 65
Projects 66

SESSION 3 STACKS FROM SCRATCH: BUILDING WITH BUTTONS 68

Objectives 68
The Problem 69
Introduction 70
Birth of a Stack 70
Stack Construction Toolbox 72
Instant Buttons, Instant Cards 83
Introducing Stackware Engineering 97
Summary 101
Key Words 101
Self-Testing Exercises 102
Projects 103

SESSION 4 HYPERCARD'S HIDDEN LAYERS: BACKGROUNDS AND BACKTRACKING 104

Objectives 104
The Problem 105
Introduction 106
Some Background on Backgrounds 106

A HyperCard Adventure 107
Stackware Engineering Revisited 132
Secondhand Art 134
Loose Ends 141
Summary 144
Key Words 144
Self-Testing Exercises 144
Projects 145

SESSION 5 CREATING HYPERTEXT: NONLINEAR WRITING 146

Objectives 146
The Problem 147
Introduction 148
Designing Hypertext 148
Making a Master Menu 154
Field Day 158
Introduction: One Page at a Time 165
Planting a Hypertext Tree 169
Unfinished Business 175
Summary 178
Key Words 178
Self-Testing Exercises 178
Projects 179

SESSION 6 BEHIND THE BUTTONS: EXPLORING HYPERTALK 182

Objectives 182
The Problem 183
Introduction 184
HyperTalk Revealed 184
Building a Dynamic Presentation 192
Moving Pictures 202
Singing Stacks 217
Smart Links 219
Missing Links 222
Summary 225
Key Words 225
Self-Testing Exercises 226
Projects 226

APPENDIX A HYPERCARD SOURCES AND RESOURCES 229

APPENDIX B PREPARING YOUR COMPUTER SYSTEM: A LAB ADMINISTRATOR'S GUIDE 239

APPENDIX C MACINTOSH SURVIVAL SKILLS 241

APPENDIX D HYPERCARD SURVIVAL SKILLS 243

APPENDIX E ASK DOCTOR HYPER: A BEGINNER'S HYPERCARD TROUBLESHOOTING GUIDE 247

APPENDIX F HYPERCARD KEYBOARD COMMANDS 251

GLOSSARY 253

INDEX 256

FOREWORD

As personal computers become easier to use—with all those screen windows, pictographic icons, and mouse pointers—they become increasingly more difficult to program. Hardy "early adopters" of personal computing had to learn and remember complex keyboard command schemes, sometimes working harder than the computer to perform a simple task. For most users today, however, getting a particular job done is the prize, not trying to figure out how to respond to a cryptic screen prompt. The environment on the screen must not get in the way of the task. Consequently, users' expectations are high. The programmer's job is enormous.

Until very recently, the formidable job of programming a personal computer dissuaded all but the most dedicated technical types. Professional and part-time hackers, nerds, and computer science students spent hours learning computer languages like BASIC, Pascal, and C to produce the Next Great Program. But every year the barrier to programming, especially in graphic environments like the Macintosh, climbed higher and higher.

Let's take a closer look at the Macintosh. When the Mac first arrived in 1984, Apple had prepared a three-volume set of programmer's manuals, called *Inside Macintosh*. These books documented all the built-in tools that were to make the programmer's job easier. That three-volume set has now grown to six, and, in hard cover, weighs something akin to a stereo component. So, if you want to program for the Macintosh in "traditional" languages, first you must master a language (Pascal or C), then *Inside Macintosh*. No wonder the software offerings for the Macintosh were so few in the first years of the Macintosh. Even the pros had a hard time of it. Not until 1986 did a respectable assortment of software tailored for the Macin-

tosh environment reach store shelves. Macintosh users (remember, they're the ones with the high expectations) had long ago rejected software products that merely added pull-down menus to existing IBM PC-style programs.

While professional programmers were learning all the tricks, Apple continued to enhance the Macintosh's powers, and the nontechnical, would-be programmer was left in the dust. Those of us especially in the arts and humanities had to rely on software developed by what I call the Programming Priesthood. I don't mean that as a pejorative term, because the priests are the ones who give us useful general purpose tools, such as word processors and spreadsheets. But there has always been need for "little" programs—special purpose programs that meet the needs of an individual or small group. Unfortunately, the wall between the machine and those with the expertise worth computerizing became insurmountable.

Until, that is, I had the rare opportunity as an Apple outsider (a free-lance writer, actually) to see an early prototype of what was to become HyperCard. John Sculley (whom I had just interviewed for *Playboy*) introduced me to HyperCard's creator, Bill Atkinson, in March of 1986. Bill showed me a few examples of applications he had made with his own new software tool—applications that I could easily create as well. I describe my reaction to this demonstration as feeling my brain expand to the size of the room and flooding rapidly with ideas.

Watching HyperCard's features solidify and seeing Dan Winkler's implementation of the HyperTalk scripting language take shape over the next several months made it clear that programming's barrier walls were to come tumblin' down. Creating screen items like scrolling text fields and clickable buttons no longer took dozens of lines of Pascal or C code—they were choices in menus and dialog boxes. Bill had already done much of the dreary work.

In addition to writing *The Complete HyperCard Handbook*, I also started working on two HyperCard-based applications that would eventually become commercial products. *Business Class* (a world traveler's guide) and *Focal Point* (a personal organizer). Within a short time, I went from an author of books and articles to software author.

I wasn't alone. Around the world, businesspeople, academicians, and students applied their expertise to HyperCard programs. People who would staunchly proclaim themselves nonprogrammers were creating multimedia training presentations, interactive cultural libraries, social science and historical simulations, student housing guides, and thousands more. HyperCard has touched thousands of lives.

As stepping stones to further computer language study, HyperCard and HyperTalk offer fine introductions to concepts of object orientation. The programming world is heading in this direction, with many professional programmers advancing to object-oriented Pascal and C++. If you should progress to these levels, you will instantly see how simple and elegant the

HyperCard environment is. Like many professionals, you may even continue to use HyperCard as a prototyping tool to test program flow and user interfaces.

Importantly, you don't have to be a computer science major to appreciate how HyperCard empowers you. If you can transform but one idea into a working HyperCard application, then consider yourself successful. It may open your eyes to even more possibilities. That is the beauty of HyperCard.

Danny Goodman

PREFACE

This book is designed to deliver what it promises: *HyperCard in a Hurry*. Rather than exhaustively describing every feature and nuance of HyperCard and HyperTalk, *HyperCard in a Hurry* systematically introduces the fundamentals of HyperCard through a series of self-study, hands-on sessions. The first two sessions introduce HyperCard as a tool for information storage and retrieval. The four remaining sessions illustrate essential techniques and tricks for building a variety of HyperCard stacks.

HyperCard in a Hurry is not *Instant HyperCard*. Learning a complex and powerful tool takes time. But by focusing on the essentials of HyperCard and stack design, *HyperCard in a Hurry* minimizes the amount of time it takes to become a competent, confident HyperCard stack designer.

After a few hours with this book, a student with little or no computer experience should be able to design, build, and customize visually impressive and conceptually complex stacks to perform a variety of tasks. This statement says as much about HyperCard as it does about this book. Before HyperCard, it simply wasn't possible for a computer neophyte, with an investment of a dozen hours, to build sophisticated graphic presentations with complex logical structures. With HyperCard, beginners become programmers every day. This book is designed to facilitate that process.

The material in this book has been tested by hundreds of students (as well as several instructors and a few innocent bystanders). Most of the students were university students in a freshman computer literacy class, where they learned how to use computers mostly through hands-on experience with Microsoft Works and HyperCard. Junior high students in a two-week summer math camp sponsored by the American Indian Science and Engineering Society provided the greatest inspiration in shaping the

book. After being led through an abridged version of the Dungeon project in Session 4, these students applied what they learned to produce visual recreations of classic Native American legends. Their creations, completed in just a few hours, should convince even the most jaded skeptic that HyperCard *can* be learned in a hurry.

This book can be used by anyone who's interested in learning HyperCard. It assumes no computer experience, but provides time-saving shortcuts and bypasses for Macintosh veterans. Each session guides students step-by-step through the process of using or creating stacks, pointing out the important concepts along the way. Projects are drawn from a variety of subject areas, so students with different backgrounds and interests can see by example how they can put HyperCard to work in practical and provocative ways.

From Zero to HyperCard in Six Sessions

Sessions 1 and 2 lay the groundwork for the rest of the book by providing a guided tour of HyperCard (with an optional introduction to Macintosh basics for rank beginners). Session 1 focuses on exploring HyperCard stacks as a browser; Session 2 shows how to use HyperCard as a personal database.

In Sessions 3 and 4, students learn the basics of stack construction in the process of creating a dynamic multilayer map/display and a complex maze game. Graphic tools, cards, buttons, links, and backgrounds are introduced gradually as these projects unfold. Aesthetic and technical design principles are illustrated and explained along the way, so students have a conceptual framework to guide their use of HyperCard's tools.

In Session 5, students create a hypertext term paper, learning how to use and manipulate text fields at the same time they're creating a complex, nonlinear document. In this session and the next, students are guided through the design process that precedes the actual computer work.

Session 6 introduces HyperTalk, the programming language that hides underneath HyperCard's friendly point-and-click interface. In this session, students learn how to use HyperTalk to add visual effects, sound effects, and animation to a presentation graphics stack. In the process, they're introduced to the basic principles of this powerful object-oriented language. With the background this session provides, students can explore scripts created by others, expanding their HyperTalk knowledge as they do.

For those students who aren't satisfied with what they learn in these six sessions, Appendix A provides links to other sources and resources, including books, software, and hardware. This section is designed for readers who want answers to questions such as "What's a good way to learn more about HyperTalk?" and "How can I put my own pictures and sounds in HyperCard?"

The remaining appendices are designed to make this book a useful reference tool. These appendices answer the most common HyperCard questions, summarize the most important HyperCard and Macintosh operations, and organize and annotate the main HyperCard menu and keyboard commands.

In the process of writing this book, I've become even more convinced of the importance of HyperCard as an educational tool. HyperCard unlocks the door that used to say, "Programmers only beyond this point!" There's an exciting world full of creative computing behind that door. *HyperCard in a Hurry* is a guidebook to that world.

ACKNOWLEDGMENTS

This book would not be what it is without the mental, physical, and spiritual contributions of many, many people. To all of you who helped me with a kind word, a thoughtful suggestion, an open ear, and/or a warm heart, I thank you—you know who you are, even if you don't find your name here. Specifically, I owe a heartfelt thank you to Frank Ruggirello, Carol Carreon, Pat Brewer, Leland Moss, James Chadwick, Anne Draus, Danny Goodman, Sarah Lillie, Brian Crissie, Rajeev Pandey, Shawn Larson, Jeannie Holmes, Mary Ann Dengler, Jeff Hino, Mike Quinn, Kevin Hurst, Pam Meyer, Ken Tubbs, Paul Ritter, Gail Sanders, Shannon Whitmore, Bryan Miller, Phil Brown, Jack Dymond, Phil (Mouse Droppings) Russell, Neal Gladstone, Chuck Sears, Don Poole, Saundra Hopkins, Mike Massey, Clara Knutsen, Bernie Feyerherm, Walter Rudd, Liza Weiman, Shoobedebop, the students and staff of the 1989 AISES Math Camp, the hundreds of CS 101 students who beta-tested this material, and the consulting staff at the OSU Computer Science Lab. I am also grateful to the following professors who reviewed the material while it was still in manuscript form: Sal Abate, Ohio State University; Dwight Barnette, Virginia Polytechnic Institute; Lee Bryant, State University of New York–Geneseo; Scot Drysdale, Dartmouth College; Joseph Fahs, Elmira College; Sherry Francis, University of Arkansas; Jim Gips, Boston College; Gordon Goodman, Rochester Institute of Technology; Annette Lamb, University of Toledo; Kurt Lauckner, Eastern Michigan University; Henry Leitner, Harvard University; Elaine Lewis, Boston University; Nate McVaugh, University of Utah; John Omohundro, State University of New York–Potsdam; Todd Papke, University of Iowa; Jeff Parker, Boston College; Lawrence Ragan, Penn State; John Slatin, University of Texas; and Ken Tuddenham, University of Utah. Most of all, thanks to my family—Ben, Johanna, and Susan—whose sacrifices and support made this book possible.

HYPERCARD IN A HURRY

INTRODUCTION

If you're *really* in a hurry to get started with HyperCard, you may safely bypass the first two sections of this introduction, skipping directly to HyperCard Essentials. But if you're totally unfamiliar with HyperCard, the material in those two sections will provide you with a background for understanding the ideas behind HyperCard.

What Is HyperCard?

HyperCard is one of the most interesting and influential pieces of software to be released in recent years. It's also one of the most difficult to describe. The question "What is HyperCard?" has several answers:

- **HyperCard is a tool for accessing information.** There's no shortage of information in today's world; the problem is finding the right piece of information when you need it. It's easy to spend hours scanning reference books for an elusive fact, quote, or idea, following footnotes and bibliographies that may lead toward other promising sources. There's no easy way to seek out particular facts quickly with conventional printed references.

  When you're looking for information with HyperCard, it's much easier to cut your own trails through the information. Unlike a conven-

tional book or article, a typical HyperCard document contains links that can lead you quickly to other parts of the document—or to other related documents. If you're researching logging practices in the Pacific Northwest, you might choose to jump from reading about the Douglas fir tree to a detailed entry on the spotted owl that inhabits that tree, perhaps clicking a button that plays a recording of the owl's call. The possibilities are nearly limitless.

- **HyperCard is a tool for managing information**. Like a standard computer database, HyperCard allows you to store all kinds of information, from phone numbers to phonetic spellings, in organized packages that can be expanded and changed to meet your needs. But because HyperCard documents can be linked, it's possible to jump instantly from a calendar reminder of a friend's birthday to a phone book that can automatically dial your friend's phone number for you.

  Many people use HyperCard simply to access "canned" information; others use it to store their own personal information in an easy-to-access form.

- **HyperCard is a software construction kit.** The Macintosh has long held a reputation for being the most user friendly of computers. With its intuitive, graphic user interface and its consistent command structure, the Macintosh has become the computer of choice for millions of people who don't want to be bothered with memorizing cryptic commands like CHDIR SYS3825 or DEL *.BAS.

  Ironically, the user friendly interface makes the Macintosh programmer-hostile when compared with other machines. Traditional Macintosh programming is incredibly difficult, in large part because the programmer has to anticipate the needs of the user in so many ways.

  HyperCard has changed all that. With HyperCard, it's possible for nonprofessional programmers to create original working software applications in hours rather than months. In the same way that MacPaint makes child's play of computer graphics, HyperCard can make a game of building software applications. Bill Atkinson, HyperCard's creator, calls it a "software erector set."

  Like any programming tool, HyperCard has strengths and weaknesses. While it's not well suited for building industrial-strength word processors or spreadsheets, it's nearly ideal for building customized personal information managers, educational tutorials, and user interfaces to other, less friendly programs.

- **HyperCard is a medium for publishing information in nonsequential form.** By simplifying the program-construction process, HyperCard has opened up the world of programming to thousands of people who consider themselves nonprogrammers. Most of these nonprofessional programmers are creating programs that are more like documents than applications. Engineers, teachers, and other profes-

sionals are using HyperCard as a medium for making their expertise available to others in a dynamic, interactive form.

- **HyperCard is a gateway to multimedia computer applications.** The user interface is often the weak link in a new piece of hardware or software. Because HyperCard shines as a user interface construction kit, it's being used to create friendly "front ends" for other software packages, hardware peripherals, and multimedia devices. With a well-constructed HyperCard front end, an otherwise complex technological device, such as a video disk player or a CD-ROM drive, can become as easy to control as a child's toy.

HyperCard, Hypertext, and Hypermedia: Beyond Books

While HyperCard is a relatively new software product, it's based on ideas that go back to the earliest days of computing. In 1945, Vannevar Bush, science advisor to President Roosevelt, wrote about his vision of a system for tracking and using scientific literature. The **memex**, as he called it, would link together articles, sketches, and photographs to form a comprehensive, cross-referenced research tool.

The technology of the time made it difficult, if not impossible, to actually create a memex. But over the years, Bush's ideas inspired Doug Engelbart, Ted Nelson, Alan Kay, and other computer visionaries to push the technology toward that end. These early efforts were generally referred to as **hypertext** because they allowed mostly textual information to be linked in nonlinear ways. Conventional text media like books are linear, or sequential: they are designed to be read from beginning to end in one particular order. Hypertext media, or **hypermedia**, invite readers to cut their own trails through information.

If you're reading about Beethoven's life, for example, you might want to learn more about his hometown, the disease that caused his deafness, the composers who most inspired him, or the culture that spawned him. If you decide to explore the culture of the time, you might be fascinated by a discussion of religious festivals and decide to focus on those. That might start you wondering about the primitive rituals that predated those festivals, and so on. With a large, all-encompassing hypertext reference, you can follow your curiosity wherever it might lead you.

You're not likely to find HyperCard stacks that give you the kind of absolute intellectual freedom hinted at here; that kind of multidimensional, cross-disciplinary information source requires an investment of time and hardware that goes beyond what most of us would consider reasonable. But because HyperCard is the first software tool to popularize the concepts of hypermedia, it's rapidly pushing us toward the visions of Vannevar Bush.

HyperCard Essentials

When you work with a Macintosh, or any computer, you use **applications**—software tools—such as word processors and graphics programs to create **documents** such as letters, research papers, and posters. In the same way, you'll use HyperCard to create documents. A HyperCard document is known as a **stack**, and it's made up of **cards**.

Like a 3×5 note card, a HyperCard card can contain any combination of graphics or text. Each card is the size of a standard Macintosh Plus or SE screen. When HyperCard is used on a large-screen Mac, a card occupies a window in the center of the screen.

It's possible to **browse** through the cards in a stack in sequential order, from the first card to the last. But in most stacks you aren't restricted to this kind of sequential access. **Buttons** on each card allow you to jump to other related cards in the stack—or to another stack—with a click of the mouse. For this kind of direct access to other cards to occur, it's necessary to **link** the buttons to those cards. But as you'll see, it's a simple process to create a button that links one card to another.

Buttons can do far more than take you to another screen. HyperCard buttons can play music, open dialog boxes, launch other Macintosh applications, rearrange information, perform menu operations, dial telephones, send messages to hardware devices; do just about anything that you can do with your Macintosh. Programming a button to perform many of these tasks requires mastery of **HyperTalk**, HyperCard's hidden programming language. But as you'll see in this book, it's possible to accomplish a surprising amount without learning a word of HyperTalk. In your last session you'll see how a few carefully chosen HyperTalk words can enhance the stacks that you create.

Charting a Path through This Book

Learning HyperCard is like learning to ride a bicycle: Reading about it just isn't enough. This book is designed to give you some hands-on experience so you can start using HyperCard productively as quickly as possible. It's divided into six sessions. Each session provides you with a new way of working with HyperCard. The more sessions you complete, the deeper your understanding of HyperCard. This kind of layered structure is possible because of HyperCard's multileveled nature. HyperCard can be just about as powerful a tool as you choose to make it.

To clarify, let's compare HyperCard to TV. When you read an instruction manual for a new TV, you learn to turn it on, adjust the volume, and navigate the channels—but not to make your own TV shows. In Session 1 of this book, you'll use HyperCard like a TV: as a medium for receiving information packages created by others. Session 2 treats HyperCard more like a

VCR; you'll learn how to capture and customize information sources. In Sessions 3, 4, and 5 you'll master HyperCard as a video camera, so you can actually create your own programs. In Session 6, you'll learn to work with HyperCard scripts, which have no analogy in the world of consumer video today. It's as if you could teach your video camera or television to do things that they couldn't do when they left the factory.

During a typical session of *HyperCard in a Hurry*, you'll use HyperCard to solve a real-world problem, learning several new concepts and techniques as you do. Each session can be completed in roughly an hour or two. (If you're a rank beginner and/or very thorough, some sessions may take longer. If you have lots of experience with Macintosh computing, you can probably zip through two sessions in the time it takes a Macintosh neophyte to complete one.)

Each session assumes you've mastered the material in the previous session, so resist the temptation to skip ahead. There may be times, though, when skipping ahead is the right thing to do. This book is designed to meet the needs of a widely diversified audience: seasoned Macintosh users, experienced programmers, and (especially) rank beginners—in short, anyone who wants to learn HyperCard in a hurry. Because different readers have different needs, this book has road signs in boxes like this to steer you around material that may be unnecessary or inappropriate for you.

In each session, you'll be asked to perform a number of tasks to complete a project. A typical task might look like this:

Resetting the User Level

The steps to perform a given task are preceded by check boxes, like this:

☐ **Point to the button labeled Cancel and click the mouse.**

Follow each instruction *carefully*, checking it off when you complete it. The paragraphs that precede and/or follow the instructions give you more details about why you're doing what you're doing. If you skip a step or don't do exactly what it tells you to do, you may take a wrong turn and stray from the path described in this guidebook.

You'll find illustrations along the way to show you what your screen should look like. Don't worry if your screen doesn't *exactly* match the ones in the book; when you're working in HyperCard's visual environment, there

are all kinds of variations that have no effect on the end result.

Even if you follow every instruction and check every screen shot, you may feel hopelessly lost at some point when things don't match. When that happens, you can either (1) panic or (2) accept your disorientation as an opportunity to do some creative detective work. If you're here to learn, go with the second approach. Retrace your steps looking for the wrong turn. Read ahead to see if you made any incorrect assumptions about where the book is leading you. Uncertainty is a wonderful catalyst for learning; make the most of it!

As you proceed through a session, don't try to memorize every step of every process. If a piece of information is crucial, this book will tell you.

> Boxes like this one contain facts, concepts, and pointers worth remembering. It's not necessary to memorize every detail in these boxes. They're easy to scan when you're reviewing a session or looking for a piece of important information later.

Like anything else, HyperCard has a vocabulary that includes terms you may not recognize. When a new term is introduced in this book, it's printed in **boldface** so it's easy to spot. At the end of each session, you'll find all of the new terms collected in a list of key words.

Exercises and projects at the end of each session will help you to test your knowledge and understanding and further develop your HyperCard skills.

If you forget a detail and can't remember where you saw it, use the reference section at the end of the book. You'll find summaries of menu and keyboard commands, a short cookbook of essential HyperCard procedures, a troubleshooting guide to answer common questions, and an annotated list of references to help you find answers to questions that aren't covered in this book.

One final word: *HyperCard in a Hurry* is intended to get you up and running with HyperCard quickly. But that doesn't mean you should rush through the material. You'll learn more and learn faster if you take your time and relax while working through these examples. HyperCard is fun—*enjoy it!*

HyperCard in a Hurry

SESSION 1

Objectives

By the end of this session you should be able to

- Use menus to perform basic Macintosh operations
- Begin a HyperCard session
- Use the Home card to locate a number of built-in HyperCard stacks
- Use HyperCard buttons to navigate a HyperCard stack and return to the Home card
- Use keyboard shortcuts to navigate through a HyperCard Help stack
- Terminate a HyperCard session

1

HYPERCARD FOR BROWSERS:

A CARD FOR EVERY TASTE

The Problem

You've contracted a serious case of information anxiety. You're sitting in the middle of an information explosion, and the gap between what you know and what you think you *should* know is getting wider every day. Traditional information-gathering tools just don't seem to help anymore. Books take too long to read, and they're often obsolete by the time they become widely available. Magazines and newspapers are timely, but the information that's in them is almost impossible to find when you need it. Radio provides no practical tool for sifting the relevant from the repetitive. And TV, which promises so much and delivers so little, is probably the worst time waster of all. Computers helped create this rapidly expanding sea of information. Can they help keep you from drowning? Perhaps HyperCard can help.

Learning to use a computer is like learning to ride a bicycle: Reading about it just isn't enough. This book is designed to give you some hands-on experience so you can start using HyperCard productively as quickly as possible. It's divided into six sessions, each of which can be completed in an hour or two. This first session will introduce you to the Macintosh and HyperCard. Whether you're a newcomer to computers or an experienced Macintosh user, you'll learn in this session how to navigate information-rich HyperCard stacks like a pro. Take your time, take notes on things that aren't clear, and take breaks when the spirit moves you. Have fun!

Customizing This Session

If You're New to the Macintosh . . .

Relax. Macintosh computing is easy, intuitive, and fun. You'll be up to speed soon if you take the time to learn a few Macintosh basics.

This book assumes that you know little or nothing about computers in general and the Macintosh in particular. But since HyperCard is the focus of this material, you won't find detailed instructions on those aspects of the Macintosh that aren't directly related to HyperCard.

If you can, start your HyperCard journey with the disk-based **Guided Tour** that Apple provides with every Macintosh. This self-paced tour is an excellent tool for learning Macintosh fundamentals. If you take the time to explore the material in this tour, HyperCard mastery will come easier and faster.

If You're a Macintosh Veteran . . .

You can safely skip the "Meet Mac" section and proceed quickly through the remaining material, starting with "Hello, Hypercard." Focus on following the boldface instructions and reminder boxes, consulting the explanations only for unfamiliar material. But beware: Some aspects of HyperCard run counter to conventional Mac wisdom. Pay attention! If you miss something along the way, feel free to go back, or simply consult the reference section at the end of this book.

Which System?

HyperCard in a Hurry is designed to work with any Mac system that will run HyperCard. But some of the details of getting started will vary slightly depending on whether your system is built around a hard disk, a network

file server, or 3.5 inch disks. (3.5-inch disks are sometimes called **diskettes**, **flexible disks**, or **floppy disks**, because the magnetic disk inside the hard plastic case is flexible.) This book will point out the most common variations. You might want to make a note of any local idiosyncrasies in the margins. Let's start with some general pointers for each class of users.

Feel free to skip those tips that don't describe the system you're using.

Hard Disk

Tips for Hard Disk Users

While HyperCard doesn't specifically demand a hard disk, it expresses a clear preference for one. An **internal hard disk** (one that's inside the Mac) or an **external hard disk** (a separate box with its own on-off switch) will save you time and frustration by allowing you to store large quantities of information in a quickly accessible form. Here are a handful of hard disk tips:

- Your hard disk should contain the standard system files, HyperCard, and the Home stack along with the standard HyperCard stacks shipped with HyperCard. If it doesn't, you'll need to have disks containing those files. See Appendix B for details about installing those files on your hard disk.
- If the hard disk is shared, remember to leave it in the same condition you found it. Don't leave it littered with your personal files, and don't change or delete any files that aren't yours.
- If you have a password-accessible partition on the disk, keep your password private.

File Server

Tips for File Server Users

While all file servers don't work the same way, most commonly used file servers simply appear as another icon on the user's screen above or below the system disk. That's appropriate, since a **file server** is little more than a giant, slow, shared disk. Like any shared property, the server probably has more restrictions than you'll find on your private property; you won't be able to erase it, for example. Most file servers require you to have a specially prepared System Folder on a disk in your disk drive (or on a hard disk). Here are some general tips:

- Make sure you have the right version of the System software for your installation.

- If your file server doesn't contain HyperCard and the Home stack along with the standard stacks shipped with HyperCard, you'll need disks containing those files.
- If you have a password-accessible storage area on the file server, keep your password private.
- Leave the server in the same condition you found it. Don't leave it littered with your personal files, and don't change or delete any files that aren't yours.

Diskette

Tips for Floppy-Disk-Based System Users

Using HyperCard from a floppy-disk-based system can be challenging, but it's not impossible. Here are a few guidelines:

- Make sure your system disk isn't cluttered with unnecessary files, fonts, and desk accessories. If you aren't sure about what should be on the disk, check with your lab administrator or consult your Macintosh manual and Appendix B of this book.
- You'll need a system disk containing the Macintosh system files, disks containing HyperCard, and the Home stack along with the standard stacks shipped with HyperCard. You'll need to swap between these disks as you work.
- If you're using high density Apple FDHD Superdrives (standard drives on all Macintoshes except the Plus since August, 1989), invest in high density (HD) disks that can hold almost twice as much information. With these disks, you can put HyperCard and the System Folder on one disk, so that you have a drive free for data files. There's one disadvantage to this method: Your disks won't be readable on Macs that don't have the Superdrive.

Tips for Everybody

- Throughout this session you'll be asked to open HyperCard stacks (documents) that are part of the basic HyperCard package distributed by Apple. If you're missing any file, simply read through the section that covers that file and continue in the next section.
- When you start modifying and creating stacks (Sessions 2 and 3), you'll need at least two blank disks: one to store your stacks and one for backups. (If you have a personal hard disk, you'll only need a disk for backups.) Back up your work regularly. (You'll learn how in Session 2.) Always save your files on at least two separate disks. For important projects, keep at least one backup in a safe place away from your others.

- If your Macintosh doesn't have Apple's FDHD Superdrive, it can't read HD disks. Be careful not to use HD disks in machines that don't have the Superdrive. (HD disks are easily identifiable; they have small square holes at two corners; standard disks have only one square hole.)
- Keep your disks away from dust, moisture, and magnetic fields. If your environment is particularly harsh, keep them in a plastic disk box, zippered plastic sandwich bags, or some other sealed container.
- Before you make major changes in a project, save a copy. Don't burn your bridges.

Meet Mac

If you're familiar with basic Macintosh operations, you may safely bypass this section and proceed to the section called "Hello, HyperCard."

Booting the Mac

It's time to make things happen. We'll start by **booting** the machine. (Why booting? The machine pulls itself up by its own bootstraps.)

☐ **Reach around the left side of the Mac and turn on the power switch about halfway down the back. (If you're using a Mac II, turn it on by pressing the key marked with a triangle in or above the top row of the keyboard.)**

The Macintosh will awaken with a chime or a beep.

☐ **If the screen is too bright or too dark, adjust the brightness with the knob located below the rainbow Apple logo (or on the right side of the Apple Mac II monitor).**

Once it's awake, the computer starts groping around for the **operating system** files that tell it how to go about its business.

If your Mac has a hard disk, it should find those files on that disk and take you straight to the desktop as soon as the drive is up to speed.

If it can't find a hard disk, it'll use the flashing disk to ask, "Where's the disk?" The Mac needs a **startup disk**—one containing **system files**. We'll refer to this disk as the system disk, but the name doesn't matter.

- **If your machine doesn't boot by itself, insert your system disk, metal end first, label up. Gently push it all the way into the slot under the screen.**

> The little metal door on the disk protects it from dangerous dust, scratches, and fingerprints, so don't open it yourself; the machine will take care of that.

You'll soon see a little Mac on the screen smiling as if to say, "Thanks." (If you accidentally insert a disk that doesn't have the system files on it, the disk drive will spit out the disk: "No thanks." If that happens, turn the machine off for a second and try again, making sure you're using a disk with system files. If it happens again, try again with a different system disk.)

After the system has finished booting, you should see your electronic **desktop** on the screen. The scene should include at least two **icons**, one representing your system disk (or the hard disk) and one labeled "Trash."

If you're connected to a **network** with a file server, your specially configured system disk should alert the server to your presence. In some cases, you may have to formally sign on by typing your name and/or a password to access the server. If you need to do more than simply turn on the machine, check with the lab staff for details.

Mousing Around

You're staring at your electronic desktop (brought to you by an important system program called the **Finder**). This desktop is your work space when you're dealing with Macintosh, and you can organize it or clutter it the way you would any desktop. But first you have to learn how to move things around. That's what the mouse is for.

- **Holding the mouse so its cable points away from you, move it around and watch the pointer move.**

You'll be rolling your mouse around whenever you use your Mac, so find a comfortable location for it and keep that area clear. If it bumps into something or reaches the edge of the desk or mouse pad, pick it up and move it to a roomier location; lifting the mouse doesn't move the pointer on the screen.

- **Move the pointer so that its tip touches the trash can icon; push and release the mouse button once.**

This is called **clicking** the mouse. Watch the trash can icon change from white to black.

- **Put the pointer on the disk icon and click the button once.**

The trash can is now back to normal, and the disk icon is now black. This inverted coloration (Apple calls it **highlighting**) indicates you have **selected** the icon by clicking on it.

> When you use the Macintosh, you generally *select* something before you do something.

- **Point to (put the pointer on) the trash can and press the mouse button. Move the mouse to the left with the button held down, and watch an outline of the can move with the pointer. Let go.**

When you let go of the button, you're letting go of the can, too, and it relocates where you left the outline. This is called **dragging** an icon. You can drag any icon this way.

- **Drag the trash can back to where you found it.**

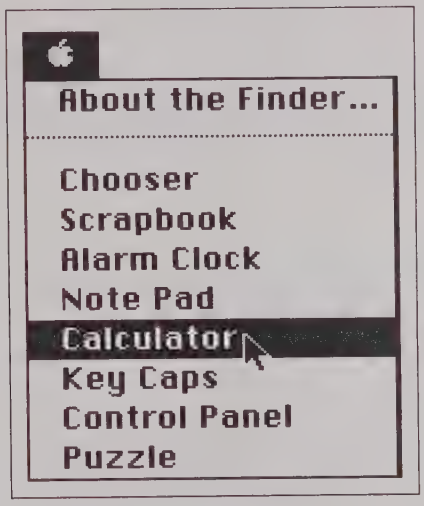
Your desk comes equipped with more than a trash can; there are also several hidden **desk accessories**. Which desk accessories you have depends on what's been included in your System file, but you probably have a calculator, an alarm clock, and a few others. These tools are available almost anytime you're using the Mac—even while you're using HyperCard.

Discovering Desk Accessories

- **Move the pointer to the in the upper left corner of the screen and press and hold the mouse button.**

The **Apple menu** that appears is called a **pull-down menu**, because you pull the pointer down to the item you want to select.

- **Without releasing the button, drag the arrow down the list. When you reach Calculator, release the button to select it.**



If you overshoot Calculator, just move the pointer back up to Calculator, or roll off the edge of the menu with the button down and try again. If you don't have a calculator in your list, pick something else.

- **Use the mouse and the keyboard to try out the Calculator.**

It works like any inexpensive calculator, except that you can use the mouse or the computer keyboard to punch the buttons. (There are three buttons that might not be obvious to you on the Calculator: the asterisk (*), the slash (/), and the C. The asterisk represents multiplication and the slash represents division. Pressing C clears the Calculator's screen and memory.)

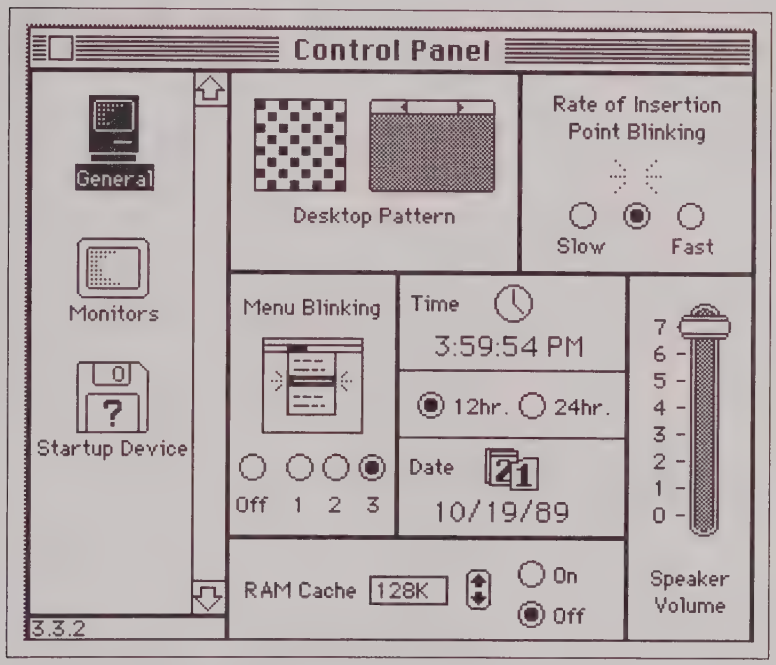
Using the Control Panel

The most important desk accessory is the **Control Panel**.

- **Select the Control Panel from the Apple menu.**

The Control Panel allows you to change everything from the desktop pattern to the sensitivity of the mouse and keyboard. Most of the time you'll have no need to change any of these settings. Today we'll check two that are important for HyperCard users: the **RAM Cache**, which controls how

much memory is reserved to cache, or temporarily store, information, and the Speaker Volume, which controls the sound output.



☐ **If the RAM Cache control has a black dot in the On button, click the Off Button.**

HyperCard on most Macs works better with the RAM Cache off. This kind of button is called a **radio button**; it's used where only one member of a group of buttons can be selected at a time.

☐ **Drag the slider for the Speaker Volume to some number greater than three; choose the setting that's most comfortable for your ears.**

(If you're using a Mac II with a color or gray-scale screen, you won't be able to see some HyperCard special effects unless you adjust your set so that it works like a standard monochrome Mac. Click on the Monitors icon in the Control Panel window. You may have to scroll to see it; scrolling is explained later in this session. Under colors, click on 2 or "Black and White." Done.)

You can drag the Control Panel around on the screen.

- **Position the mouse pointer on the Control Panel's title bar (at the top, where the name is), press the button, and drag it to one side so that the calculator is visible.**

  Notice that the Calculator's title bar is no longer black; that's because the Calculator is not **active**. When you opened the Control Panel, it became the active object on the screen.

- **Click on the Calculator to make it active.**

  Now the Control Panel moves to the background, since you've selected the calculator.

- **Select (by dragging) Close from the File menu.**

  Since the Calculator is the selected object, you're commanding it to close. It should vanish from the desktop, to await your next summons.

- **Close the Control Panel by clicking in the close box at its upper left corner.**

  This is the shortcut for selecting **Close** from the **File** menu.

Opening Windows

A **window** is a tool for looking into something. Most of the icons on your desk have windows associated with them.

- **Click on the system disk (or hard disk) icon and select Open from the File menu to open the disk window.**

  Notice that some of the commands in the File menu are gray; those are commands that aren't appropriate for what you have selected; the Macintosh dims them so they're unavailable. When you select Open, the window that zooms out from the disk icon should display at least one icon: the **System Folder**, which contains the system files.

- **Position the mouse pointer on the System Folder and click twice in rapid succession (double-click).**

  Double-clicking is the shorthand way to open something — in this case, a window looking into the selected folder. (If nothing changed on the

screen, try again, making sure that the two clicks are close together.) As you can see, there are many icons in the System Folder, each representing one of the files used by the operating system to keep things running smoothly behind the scenes.

Changing the View in a Window

☐ **If you see a list of file names rather than a collection of icons, select By Icon from the View menu.**

> The **View menu** allows you to look at the information in a disk or folder window in several different ways, depending on whether you want information about the size, most recent modification date, or type of the files. If you don't need this information, it's usually easier to work with files in the icon view.

Notice that there's an icon count below the title bar. If the count doesn't match what you see, it's because some of the icons are hidden outside the borders of the window. You can easily hide more by resizing the window.

☐ **Click in the size box in the lower right corner of the window and drag it to a point about two inches below and to the right of the window's close box (upper left corner).**

It's hard to see the big picture through this compact window, but it's possible to see everything by using the **scroll bars**— those arrow-ended bars on the right and bottom edges of the window.

☐ **Point to the up-arrow and press the mouse button. Then point to the down-arrow and press.**

Your view will scroll up and down through the window, with the white **scroll box** moving through the scroll bar like a tiny elevator to show you the position of your window relative to the overall listing.

☐ **Try dragging the scroll box to a different position on the scroll bar. Then try clicking in the gray area above and below the box.**

These kinds of scroll bars are common throughout all Macintosh applications; they always work the same way.

- ☐ **Click in the zoom box in the upper right corner of the window.**

This box toggles the window between its current size and a larger size that almost fills the screen.

Knowing how to work with folders and windows is important when you're trying to locate and organize HyperCard stacks. But you normally don't need to look in the System Folder, so let's close it up and open HyperCard.

Closing Windows

- ☐ **Close each open window on the screen using either the close box or the Close command in the File menu.**

If you've worked with other computers, you may be surprised to learn that you have to ask the Macintosh for your disk before you can take it out of the drive. This exercise in politeness prevents you from accidentally damaging your disk by removing it at an inappropriate time.

- ☐ **Click on the system disk icon and select Eject from the File menu (unless you're working from a hard disk).**

(Remember the rule: Select, then do.) The disk should pop out like a piece of toast. (This won't work if you're using a hard disk because the hard disk spins continually.) The disk icon is still on the screen, but it's now shaded gray—gone but not forgotten.

- ☐ **If you have two drives, re-insert the disk.**

The computer will sometimes need to refer to this disk, so if you're using a one-drive system, you will be asked to swap this disk in occasionally.

Hello, HyperCard

Launching HyperCard

- ☐ **If HyperCard is not on a hard disk or file server, insert the HyperCard & Stacks disk (or any disk containing HyperCard and the Home stack).**
- ☐ **Open the window for the disk or folder that contains Home by double-clicking (clicking twice in rapid succession) on the icon.**

☐ **Locate the Home icon on the disk or in a folder.**

Home may be hidden inside a folder. Try opening any likely-looking folders. You may find folders inside other folders; just keep snooping until you locate it.

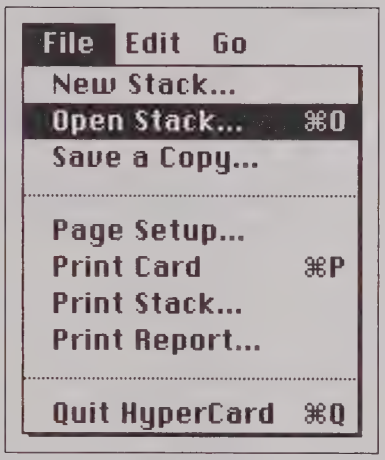
☐ **Double-click on the icon to open Home. (You could also select the Home icon and select Open from the File menu.)**

When you launch HyperCard, you're transported to the **Home** card — your HyperCard base of operations. Your Home card should look something like this:

(If you have more menus at the top of your screen than Apple, File, Edit, and Go, just ignore them for now.)

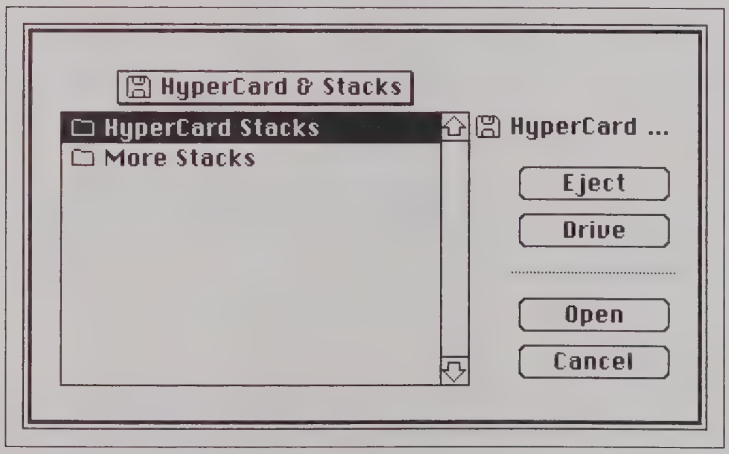
Opening via the File Menu

HyperCard has pull-down menus just like the ones on the Finder's desktop. Virtually every Macintosh application has similar menus. This kind of consistency makes it easy to apply what you know about one tool (such as a word processor) when you learn how to use a new one (such as a graphics application). Let's take a minute to try one of the commands to see how it differs from its desktop cousin.



☐ **Select Open Stack... from the File menu.**

You'll see a **dialog box** like this one. Dialog boxes are, as the name implies, used to establish two-way communication between you and the computer. Sometimes you respond to a dialog box by typing something, but more often you respond simply by clicking on-screen buttons.



This particular dialog box contains a list of HyperCard files and folders that are visible on the current disk. (It won't show files that can't be opened by HyperCard, but other than that, it shows the same files and folders you'd see from the desktop.) The **Drive** and **Eject buttons** allow you to switch views to another disk drive and eject the currently selected disk, respec-

tively. Using a combination of these two buttons, you can access any disk that's readable by your machine.

We're going to open the Slide Show stack. This stack is generally located in the More Stacks folder on the HyperCard & Stacks disk. (If you're using a hard disk, file server, or specially customized disk, you'll need to find the stack by applying the principles, rather than the letter, of these instructions, because the folders may not be organized quite the same way.)

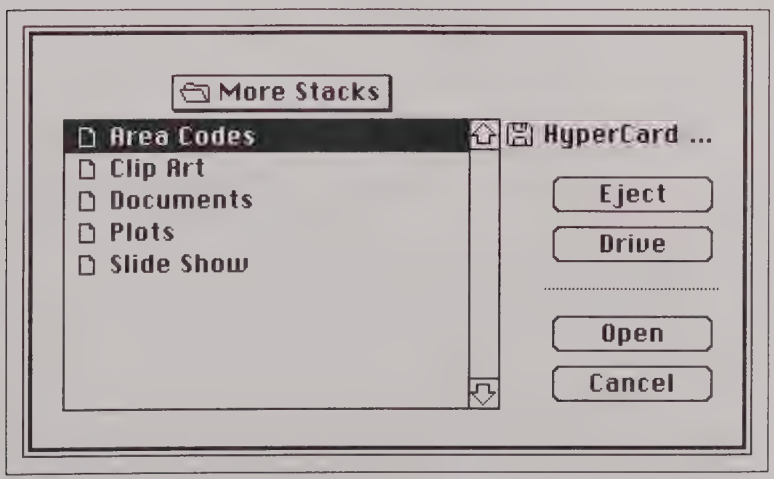
- ☐ **If you're working with a hard disk or file server that contains a HyperCard & Stacks folder or a More Stacks folder, or some other folder containing the standard HyperCard stacks, click on the Drive button until you see that disk or server in the dialog box. Then double-click on the appropriate folder to open it.**

- ☐ **If you don't have the Slide Show stack on a hard disk or server, you'll probably need to use the HyperCard & Stacks disk. Click Drive until the Eject button is not dimmed, click Eject, and insert the disk. (If you have a vacant disk drive, there's no need to eject a disk; simply insert the disk in the vacant drive.)**

Locating a File in a Folder

- ☐ **Open the More Stacks folder in the dialog box by double-clicking on the tiny icon labeled More Stacks.**

That folder's icon replaces the disk in the dialog box, and the files stored in that folder are listed.



- **Double click on the icon called Slide Show. (You may, if you prefer, click once on the file and click on the Open button.)**

When a picture fills the screen, you've successfully opened the Slide Show stack. This stack shows how HyperCard can be used to display pictures.

- **Click anywhere on the picture to make a new picture appear. Repeat until you're transported back to the Home card.**

(You'll learn in Session 6 how to create a stack like this one, complete with dissolve video effects.)

As you can see, the Open Stack... command works like the desktop Open... command, except that you tell it what you want to open after, rather than before, you select the command. Let's try it once more.

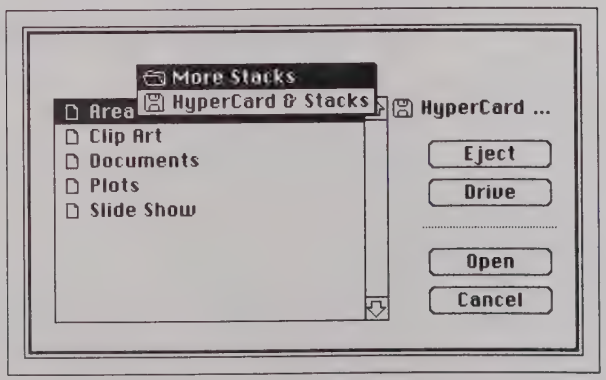
- **Select Open Stack... from the File menu.**

You'll see the same dialog box. The small icon above the list of files tells you where you are.

- **If that icon isn't the More Stacks folder, retrace your steps so you're back inside that folder.**

You really don't want to open anything in this folder; you're here to learn how to navigate your way out of a folder in a dialog box. So for the sake of learning, assume you want to open a file that's not in this folder.

- **Point to the More Stacks folder icon on the left side of the dialog box and press the mouse button.**

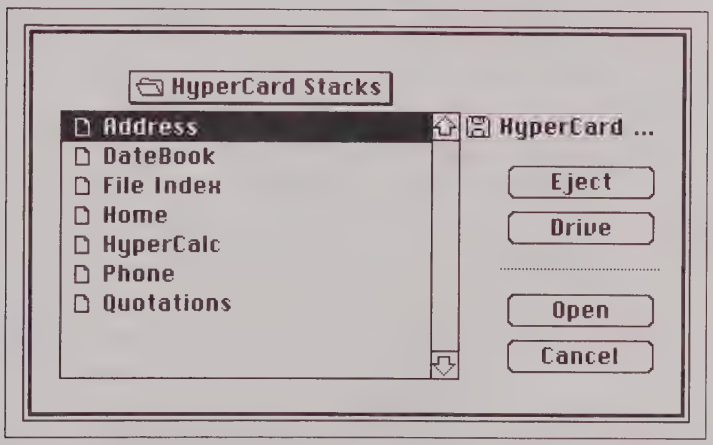


The folder turns into a **pop-up menu**, with each menu item representing one level of the folder hierarchy. In this case, there are no folders nested within folders, so the menu only has two items: the More Stacks folder and the HyperCard & Stacks disk.

- **Drag down this menu to the HyperCard & Stacks icon and release the button.**

You're looking once again at the two folders in the HyperCard & Stacks disk. There are no stacks to open at this level, so all we see in the dialog box are folders.

- **Double-click on the HyperCard Stacks folder.**



You're inside the other folder now. We could open any of the listed stacks with a double-click, but let's not.

- **Click on the Cancel button in the dialog box to retract the Open Stack. . . command and return Home.**

> The **Cancel button** in a dialog box works like an emergency exit. When you click the Cancel button, you're essentially saying, "Never mind."

These are the kinds of dialog boxes you'll see repeatedly when you're working with the **Macintosh's Hierarchical File System (HFS)**.

Where is ____?

When you're working through this and future sessions, you may see a dialog box that asks you "Where is XYZ?" where XYZ is the name of a particular HyperCard stack. Except for that question, this dialog box looks and works like the Open Stack... dialog box. Respond by locating the missing file, remembering the HFS dialog box navigation rules.

Here is a summary of those rules:

HFS Dialog Box Navigation Rules

- Clicking the Drive button switches to the next drive (or file server). Repeatedly clicking Drive cycles through all available drives.
- Clicking the Eject button ejects the disk currently displayed in the box. (This button is dimmed if the current drive is a file server or hard disk.) It's useful if you need to insert another disk.
- Double-clicking a folder opens that folder and takes you into that folder.
- Dragging down from the current folder icon reveals a pop-up menu that takes you out of that folder.

As you can see, it's not difficult to move in and out of folders when you're looking for a document. But folder hopping gets old in a hurry if documents aren't stored according to some kind of sensible organization scheme. Keep that in mind when you're creating and organizing stacks in later sessions.

It's comforting to know that HyperCard menus work like all Macintosh menus; that kind of consistency makes it easy for Macintosh veterans to quickly master new software packages. But buttons, rather than menus, are HyperCard's primary navigation tools.

Taking a Quick Tour

The Home card is populated by a variety of icons. Each of these icons is a HyperCard **button** waiting to perform some operation when you click it with a mouse.

Intro

☐ **Click once on the icon/button labeled Intro.**

HyperCard buttons, like all Macintosh buttons, respond when clicked *once* with the mouse. This is always true—even when the buttons look like icons.

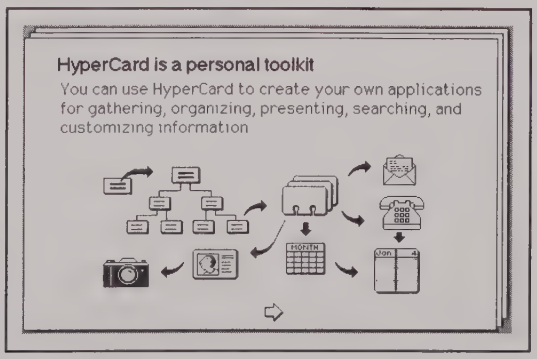
This click transports you to another HyperCard stack that will introduce you quickly to the basics of HyperCard: stacks, cards, buttons, and links.

(If you see a dialog box that says "Where is Help?" locate the Help file using the folder icons and the Drive button. If you can't find the Help file, you won't be able to use the Intro tour, since it's part of that file.)

The menu bar is gone, but there are two buttons on this screen: "Go back" and "What is HyperCard?"

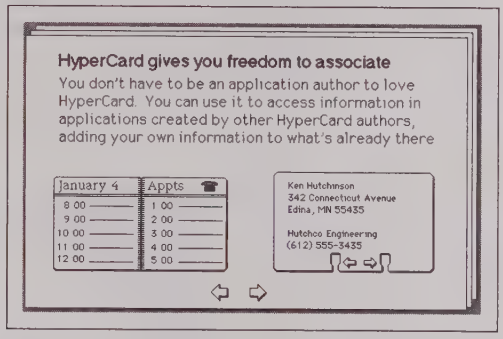
- **Click on the button that says "What is HyperCard?"**

This screen will appear:


HyperCard is a personal toolkit
You can use HyperCard to create your own applications for gathering, organizing, presenting, searching, and customizing information
MONTH
Jan

The screen is designed to resemble a stack of cards. The **right-arrow** at the bottom of the card is another HyperCard button. This kind of button has a standard meaning in HyperCard: Go to the next card in the stack.

- ☐ **Click on the right-arrow button after you've read the card.**

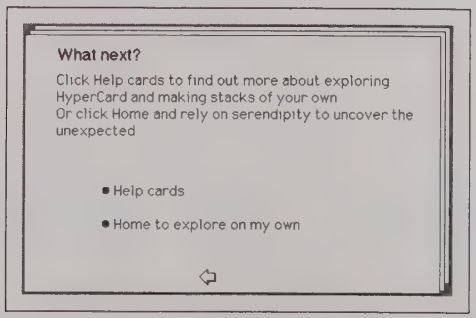


The next card has two arrow buttons: one for moving forward through the stack, and another for going back to the previous card.

- ☐ **Click on the left-arrow button to go back one card.**
- ☐ **Click on the right-arrow button to return to this card.**
- ☐ **Click on the right-arrow button after you've read the information and followed the instructions on each card.**

As you proceed through this short introduction, you'll see that HyperCard buttons can take a variety of shapes and perform a variety of operations.

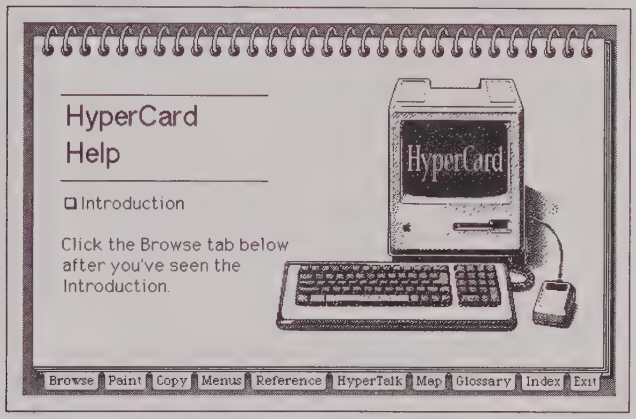
- ☐ **Proceed through the stack until you see this card:**


What next?
Click Help cards to find out more about exploring HyperCard and making stacks of your own
Or click Home and rely on serendipity to uncover the unexpected
• Help cards
• Home to explore on my own

This card actually contains three buttons: the clearly visible left-arrow button to take you back one card and two **hidden buttons** that will transport you back Home or to the HyperCard Help stack.

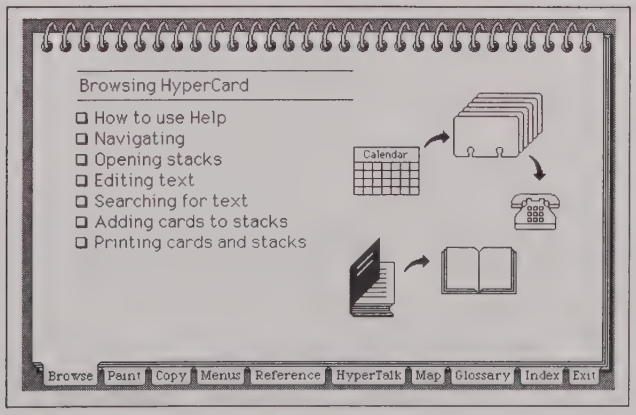
Asking for Help

- **Click anywhere on the words "Help cards."**



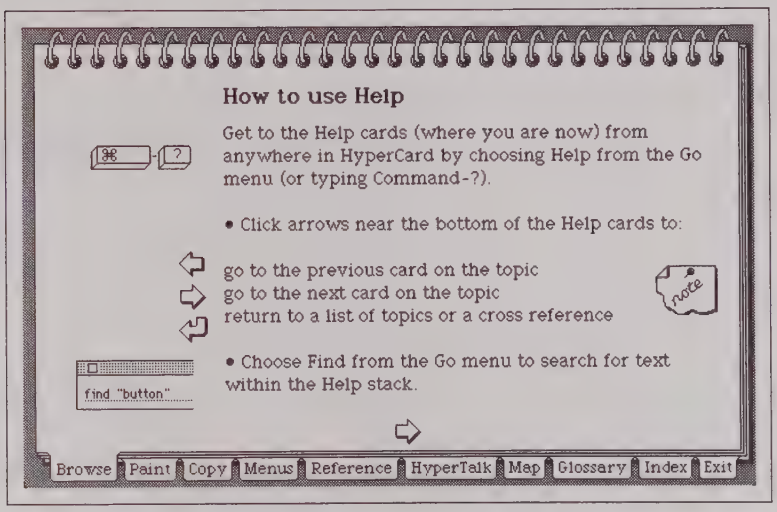
The HyperCard **Help stack** is designed to look like an index-tab spiral notebook. Each of the tabs is a button linked to a set of cards on the named subject.

- **Click on the Browse tab (bottom left) and explore the Browse help cards.**



You've opened the Help "notebook" to the section entitled "Browsing HyperCard." Several topics are covered in this section.

- ☐ **Click on the first topic, "How to use Help."**


How to use Help
Get to the Help cards (where you are now) from anywhere in HyperCard by choosing Help from the Go menu (or typing Command-?).
• Click arrows near the bottom of the Help cards to:
go to the previous card on the topic
go to the next card on the topic
return to a list of topics or a cross reference
note
find "button"
• Choose Find from the Go menu to search for text within the Help stack.
Browse
Paint
Copy
Menus
Reference
HyperTalk
Map
Glossary
Index
Exit

This two-page section describes the basics of navigating the Help stack.

- ☐ **Click on the tacked-on note button; when you've read the pop-up note, click OK.**
- ☐ **Click on the right-arrow at the bottom of the first card to turn to the next page.**

This page has no right-arrow because it's the last page in the section. It has a left-arrow button for going back one page and a return-arrow button for returning to the title page of the section.

The **return-arrow button** always takes you back, but how far back depends on what you're doing. It might take you back to the stack from which you came; it might take you back to a title page; it might take you back to a higher level card containing a menu, as it does here.

- ☐ **Click on the return-arrow button.**

In this session we cover the first three topics listed on this Help card: how to use Help, navigating, and opening stacks. We'll cover the remaining browse topics in the next session. Refer to the Help stack anytime you want to reinforce or clarify what you learn here.

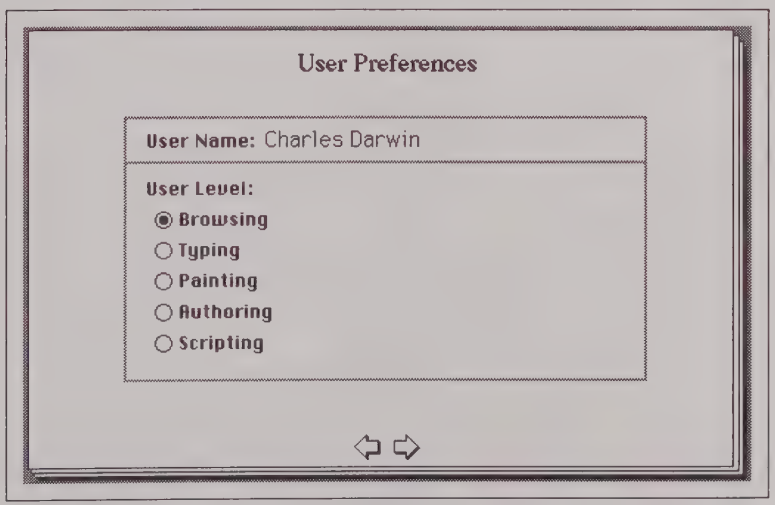
- ☐ **Click on the Exit tab at the bottom right of the Help notebook to return to Home.**

Resetting the User Level

Notice that the Home card is designed to look like the top card in a stack. In fact, it *is* the top card in the HyperCard Home stack. There should be four other cards in the Home stack. You can use the arrow buttons at the bottom of the card to go forward and backward through these cards. As usual, the right-arrow button takes you to the next card in the stack, while the left-arrow button takes you to the previous card in the stack. What happens if you try to go to the previous card when you're at the top of the stack?

- ☐ **Click the left-arrow button.**

Since there's nothing before this card, the arrow takes you around to the bottom card of the stack, which should look something like this:



In addition to having a space for the user name (which we can ignore for now), this card allows you to set the **user level** to any one of five settings, depending on how much freedom and responsibility you want to assume

when working with HyperCard. For now, all we'll be doing is **browsing** through existing stacks. To avoid the possibility of accidentally damaging or deleting any important information while you're learning about HyperCard, you can set the user level so that it allows you to do *only* browsing.

- ☐ **Click on the Browsing button.**

When you're ready to start modifying stacks and creating your own, we'll return to this card and reset the user level so you can do more than just look. But for now, let's go back Home.

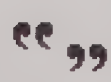
- ☐ **Click on the right-arrow button.**

This normally moves you forward through the stack, but since you're already at the end, it wraps you around to the beginning: Home.

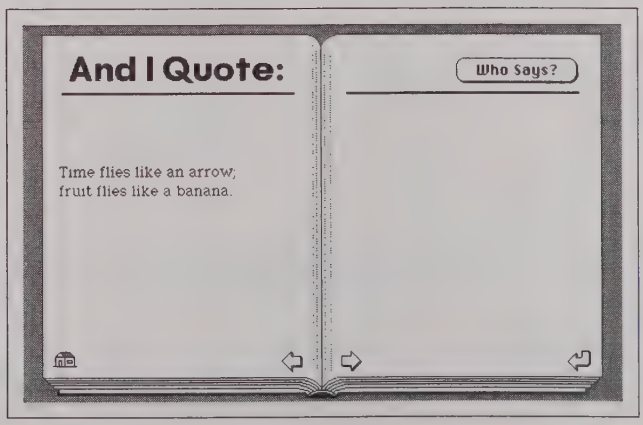
HyperCard Navigation Tips

Browsing with Buttons

The Home card is populated with a number of interesting icons. Each of these icons is actually a button that will take you to a particular HyperCard stack when you click once on it. You've already seen what happens when you click on the Intro icon. As you might have guessed, the Help icon takes you to the HyperCard Help stack. (Try it, if you like; then return Home by clicking Exit, as before.) We'll tour a few of these sample stacks and in the process learn the basics of navigating in HyperCard.



- ☐ **Click on the Quotations icon.**



Each card in this Quotations stack has a quotation and five buttons: one to tell you the author of the quote, one to take you to the next page, one to go back one page, one to take you back where you came from, and one to go Home.

☐ **Experiment with the different buttons in this stack until you understand what each does. Then click on the Home button to return to the Home card.**

You may have noticed that the return arrow and the Home icon on this card appear to do the same thing. In this case, they do: Both take you back Home. But they can have very different results, depending on the situation.

> The *Home button* always takes you Home. The *return-arrow button* takes you back where you came from, whether it's Home or somewhere else.

It may seem subtle now, but the extra bit of intelligence that's built into the return-arrow button makes it much more versatile in many HyperCard situations.

There are some other standard HyperCard icons that are helpful for serious browsers.

☐ **Click on the Clip Art icon.**

This stack is full of **clip art** that you can use when you create your own stacks. We could use the left- and right-arrow buttons to browse through the cards in this stack, but we're in a hurry.

☐ **Click on the icon that looks like three flying cards.**

You just told HyperCard to show you all of the cards in this stack very quickly.

☐ **Click on the return arrow to return Home.**

☐ **Click on the Art Ideas icon.**

This stack is also full of art that you can use when you create your own stacks.

☐ **Click on the icon that looks like a cartoon dialog balloon.**

The cartoon balloon icon, called an **about... button**, generally means "Click here to find out about this stack or card."

- ☐ **Click on the Index button to return to the first card of the Art Ideas stack.**

You can use arrow buttons to browse through this stack, or you can search more systematically using the index.

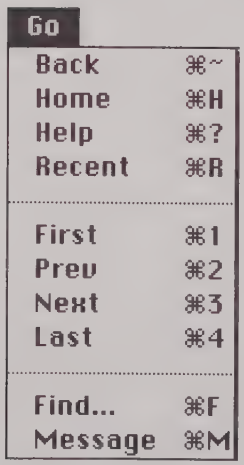
- ☐ **Use the stack index to locate different art ideas. Return to the index after each selection. (Note: The index has a second page.)**
- ☐ **Click on Home when you're through.**

Navigating by Keyboard

The Intro stack told you about a special key: the tilde (~) or Esc key in the upper left corner of the keyboard. This key takes you back one step in your HyperCard journey each time you press it.

- ☐ **Press the tilde (or Esc) key to go back one card. Press it several times, observing where it takes you. Continue to press it until it takes you out of the Art Ideas stack, through home, and into the Clip Art stack.**

Pressing the **Go Back** key (tilde or Esc) retraces your steps through the cards, reversing the order in which you viewed them. This is *not* the same as going backward through the stack.



When you're done exploring, it's time to go Home. But instead of clicking on the Home button, let's use the keyboard.

Look at the Go menu, and notice the symbols to the right of each command. Those symbols represent **keyboard shortcuts** for those commands. The key with the cloverleaf symbol (⌘) is called the Command key, because it allows the other keys to represent commands rather than characters. For example, **Command-H** is the keyboard shortcut for **Go Home**.

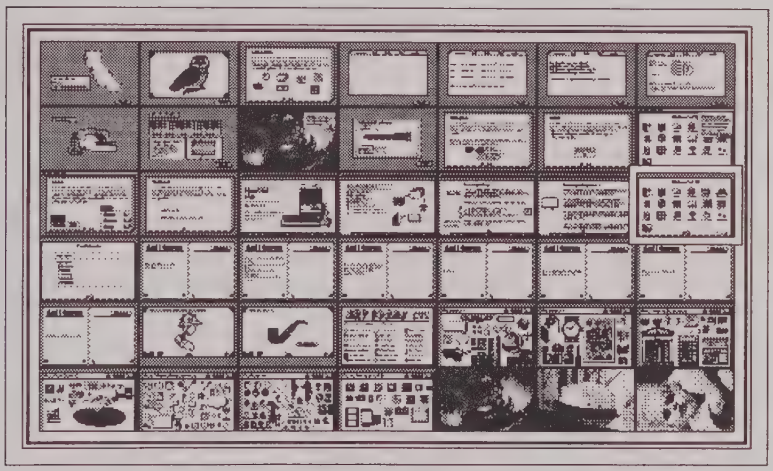
- ☐ **Hold down the Command (⌘) key and press the H key.**

This has *exactly* the same effect as clicking on the Home button. But this route Home is open even when there's no Home button on the current card.

- [ ] **Hold down the Command (⌘) key and press the question mark (?) key: Command-?**

This is the keyboard shortcut to HyperCard Help.Here's another particularly useful Go command with a keyboard shortcut:

- [ ] **Type Command-R (Go Recent).**



You're looking at a series of tiny pictures of HyperCard screens that you've seen. It's as if you took one snapshot each time you visited a card for the first time. Those snapshots are arranged here in the order you took them.

> Go Recent (Command-R) displays the last 42 cards of the session in the order they were visited; not necessarily in the order they appear in the stack(s).

- [ ] **Click on any of the small cards to go to that card; from that card, go Home.**

Let's browse our way through the Home stack by using the keyboard shortcut for the Go Next command.

- [ ] **Type Command-3 repeatedly until you return to the Home card.**

This command has the same effect as clicking on the right-arrow buttons, but it works even if the current card has no right-arrow buttons.

☐ **Type Command-2 (Go Prev) repeatedly until you return to the Home card.**

This has just the opposite effect; it goes backward through the stack, one card at a time, wrapping around to the back when it reaches the first card.

☐ **Type Command-4 (Go Last).**

This takes you to the last card in the stack.

☐ **Type Command-1 (Go First).**

You can also use the **arrow keys** on your keyboard (if you have them) to navigate. The left- and right-arrow keys work the same as left- and right-arrow buttons on the screen; the down-arrow is another Go Back key (like ~), and the up-arrow retraces your steps forward through cards you've gone back through with the down-arrow.

☐ **Try navigating through this stack using the arrow keys. End your explorations on the Home card.**

Time for two more quick tricks.

Hiding and Showing the Menu Bar

☐ **Press the Command key and the Space bar at the same time.**

Notice how the menu bar disappeared.

☐ **Type Command-Space again.**

The menu bar is back. When you're learning HyperCard, it's best to keep that bar visible. But because many stacks hide the menu bar without your permission, it's important to know how to get it back when you need it.

Finding Hidden Buttons

As you've seen, not every button has an icon. How can you know where buttons are hiding on a card? There's one sure-fire technique for exposing hidden buttons.

☐ **Hold down the Command and Option keys at the same time (Command-Option).**

Every button on the card is outlined as long as you hold those two keys down. This is an important trick to remember when you're working with stacks that don't use icons for buttons.

There are many other keyboard commands in HyperCard. Here's a summary of the HyperCard keyboard commands you've seen so far:

Keyboard Commands (and Their Menu Equivalents) for Browsers

Go to the last card visited. Go forward through retraced cards.	(Command)-~ or down-arrow up-arrow	Go Back
Go to the previous card in the stack	Command-2 or left-arrow	Go Prev
Go to the next card in this stack.	Command-3 or right-arrow	Go Next
Go to the first card in the stack.	Command-1 or up-arrow	Go First
Go to the last card in this stack.	Command-4 or down-arrow	Go Last
Go to the Help stack.	Command-?	Go Help
Go to the Home stack.	Command-H	Go Home
Show most recent 42 cards visited.	Command-R	Go Recent
Open a new stack.	Command-O	Open Stack...
Show outlines of all buttons.	Command-Option	
Show or hide the menu bar.	Command-Space	
Quit HyperCard, return to Finder.	Command-Q	Quit HyperCard

As you can see, there are many ways to accomplish the same task in HyperCard. It's not necessary to memorize all the different commands, shortcuts, and tricks to make the most of HyperCard as a tool. Use the ones that work best for you, adding new tricks to your repertoire as you need them.

Ending a HyperCard Session

- ☐ **Type Command-Q (or Select Quit HyperCard from the File menu) to return to the desktop.**

Shutting Down

If you just turn off your machine when you're through, you won't be able to take your disk with you. What's worse, you won't give the machine a chance to take care of last minute disk-keeping details.

- ☐ **If someone is waiting to use your machine, select Restart from the Special menu and leave it on. Otherwise, select Shut Down and turn it off.**

Summary

After just one session, you're well on your way to HyperCard literacy. If you're new to the Macintosh, you learned how to start your computer, move things around on the desktop, issue commands via pull-down menus, use desk accessories, work with windows, and end a Macintosh session. You learned how to launch HyperCard and navigate through stacks using on-screen buttons, menu commands, and keyboard shortcuts. You've seen how cards in HyperCard stacks can be linked in a variety of ways, and you've tried buttons that performed a variety of functions.

Session 2 will show you how HyperCard stacks can be used to keep track of all kinds of text and numerical information. You'll learn how to store, organize, and retrieve that information efficiently using a set of cleverly linked personal information management stacks. Then the real fun begins: In Session 3, you'll learn how to build your own HyperCard stacks.

Key Words

about... button
active (window or object)
Apple menu
arrow keys
booting
browsing (user level)
button (dialog box and HyperCard)
Cancel button (dialog box)
clicking
clip art
close box (window)
Close command
Command key
Command-Option (show buttons)
Command-Space (show/hide menu bar)
Control Panel
desk accessories
desktop
dialog box
double-clicking
dragging
Drive button (dialog box)
Eject button (dialog box)
Eject command (Command-E, Finder
File menu
file server
Finder
floppy disk (diskette)
folder
Go Back (~ or Command-~)
Go First (Command-1)
Go Help (Command-?)
Go Home (Command-H)
Go Last (Command-4)
Go Next (Command-3)
Go Prev (Command-2)
Go Recent (Command-R)
Guided Tour
hard disk (internal and external)
Help stack
hidden button
Hierarchical File System (HFS)
high density (HD) disk
highlighting
Home
Home button

HyperCard
icon
keyboard shortcuts
left-arrow (button)
network
Open Stack command (Command-O)
Open command (Command-O, Finder)
operating system
pointer
pop-up menu
pull-down menu
Quit HyperCard command (Command-Q)
RAM Cache
Restart command (Finder)
return arrow (button)
right-arrow (button)
scroll bar
scroll box
select
show all cards button
Shut Down command (Finder)
size box (window)
Special menu (Finder)
stack
startup disk
system disk
System Folder and system files
title bar (window)

Self-Testing Exercises

1. What is the difference between the Go Back command and the Go Prev command? Under what circumstances will they produce different results?
2. What is the difference between the Go Next command and the right-arrow button? Under what circumstances will they produce different results?
3. List several different things that can happen when you click a button in HyperCard.
4. What happens when you double-click a button in HyperCard?
5. List three ways to go to the next card in a stack.
6. What is the standard meaning for each of these icons in HyperCard?

a.

b.

c.

d.

e.

f.

SESSION 2

Objectives

By the end of this session you should be able to

- Perform basic file and disk management operations on the Macintosh
- Use a collection of HyperCard stacks to manage your personal information
- Enter text and numeric information into HyperCard fields
- Edit text in a field
- Use the Clipboard to copy and move text from one field to another
- Add a card to a stack
- Delete a card from a stack
- Sort cards into alphabetical order
- Use the Find command to locate a card with a specific piece of information

2

THE DYNAMIC FILE CABINET:

INFORMATION STORAGE AND RETRIEVAL

The Problem

Your work space is a mess. You haven't paid your bills for two months because you can't find them on your desk. You need to call your brother, but you lost the slip of paper that contains his new phone number. You have an appointment with the head of your department sometime next week, but you can't remember where you wrote down the time. You're within two days of the final deadline for your history term project, and your notes are scattered everywhere. If only you could organize all these bits of information in your life. Perhaps HyperCard can help.

In Session 1 you explored HyperCard as a window shopper, browsing but never touching or changing anything. Now it's time to learn how to personalize some HyperCard stacks by adding new information, changing old information, and rearranging existing information. HyperCard comes with several practical stacks that work together to provide a personal information management system. We'll use those and others to put HyperCard through its paces as an information storage and retrieval system.

HyperCard as a Database

Many of the tasks we'll be asking HyperCard to do in this session would normally be done in the business world by a **database** program — a program designed for the organized storage and retrieval of information. In a typical database, a **data file** (like the library's card catalog) is made up of many identically structured **records** (the cards in that catalog), each of which is divided into **fields** of information (author's name, book title, and so on).

At first glance, HyperCard has a similar structure: Stacks are made up of cards, each of which can have fields. In fact, this similarity is no accident; HyperCard performs many common database functions with ease. But as you'll see in this session, HyperCard's buttons allow it to be used in ways that you wouldn't think of using a conventional database.

Protecting the HyperCard Ecosystem

In this session you'll be working with and modifying existing stacks. Because HyperCard *automatically saves changes regularly throughout a session*, those changes are, effectively, permanent. If you're working in a public lab using shared copies of HyperCard and its stacks, it's important that you remember that others will be following in your footsteps. The procedures you'll follow in this session are designed to leave all of the stacks pretty much the way you found them, so the next HyperCard user can have equal access.

Nonetheless, mistakes happen. If you're in a public lab, chances are the lab administrator keeps a **master copy** of all standard HyperCard stacks so that modified copies can be replaced as needed. If you're working with your own system, you should probably follow that lead and make **working copies** of the files you'll be changing so that the originals remain unaltered.

Whether you're working with public files or not, it's important to know how to make **backup copies** of your files. Whenever you create something with a computer, there's a risk that you'll lose what you've done when your hard disk bites the dust or your floppy disk goes through the laundry in

your shirt pocket. You can protect yourself against that kind of calamity by regularly backing up your stacks on different disks.

The next section will lead you through the process of organizing the stacks you'll need for this session on a single disk. If you use this disk as your working disk throughout this and future sessions, you'll avoid the possibility of messing up shared public files. (If you're using a *personal* hard disk system with no shared files, you can use this as a backup disk and put a working copy on your hard disk.)

In the process of creating working and backup disks, you'll learn the basic Macintosh disk and file procedures so you can continue to keep your HyperCard stacks in well-organized folders on personal working and backup disks. You'll need two blank disks in addition to your system disk or hard disk to complete this section.

Stacks, Macs, and Backups

This section has two goals: to familiarize you with Macintosh file operations, and to prepare a disk containing the files you'll use throughout the rest of the session. If you're familiar with the Macintosh operations for copying, moving, and organizing files, you may want to skip this section and simply prepare the disk on your own. At the end of this section, under the heading "Customizing your Work Space," you'll find a description of the finished disk, along with instructions for how to use it with your particular system. If you have trouble configuring the required disk, return to this point and let this section guide you there.

Getting Started Again

- ☐ **Turn on your machine. If you're using a hard disk, the machine will boot automatically. If you're not using a hard disk, you'll need to insert your personal system disk before the machine will boot, as explained in Session 1.**

Initializing a Disk

Let's start by initializing one of your blank disks. Why? A new disk contains no information at all—just a magnetic surface waiting to be used. But your disk drive won't let you store any files on it until you **initialize** (or **format**) the disk. If you think of the disk as a freshly-paved circular parking lot for

files, initializing the disk is like painting stripes and stall numbers on the pavement so the attendant can easily park and unpark the files.

- ☐ **If you have only one disk drive and that drive is occupied by your system disk, select Eject from the File menu (or type Command-E).**

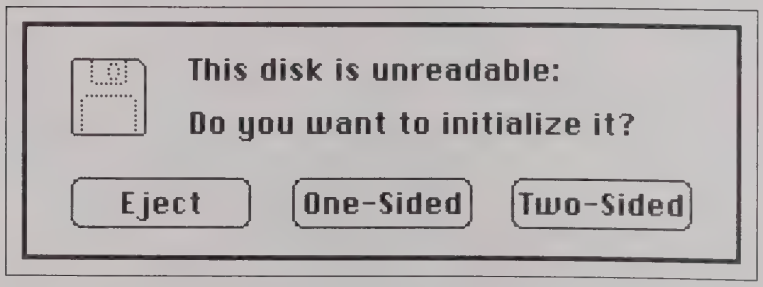
That will free the drive for your blank disk; it will probably mean that you'll have to swap these disks in the drive as you work. The Mac will prompt you whenever that's necessary.

- ☐ **Locate one of your blank disks and make sure it's unlocked.**

Check the little square hole in the corner to the right of the label. If you can see through the hole, the disk is **locked (write-protected)** so the computer can't add to, change, or erase any of the information stored on it. You'll need to slide the plastic cover over the hole if it's not already there.

- ☐ **Insert the blank disk in the available disk drive.**
- ☐ **If you are recycling a disk that's been used before in a Macintosh, an icon representing that disk will appear on the desktop. If that's the case, click on it to select it and choose Erase Disk from the Special menu. (Are you sure there's nothing of value on this disk? There's no turning back!)**

The Erase Disk command tells the computer to reinitialize the disk. Whether you're initializing a new disk or erasing a used disk, you'll see a dialog box something like this:



If you're given a choice, you'll want to select two-sided so that you can store twice as much information (800K vs. 400K) on the disk. (One K is approximately 1000 characters worth of storage space.) There's one disadvantage to two-sided disks: They can't be used in a pre-1987 single-sided Macintosh drive. If your lab has a mix of one- and two-sided drives, *be*

careful: It's easy to wipe out everything on a double-sided disk by putting it in a single-sided drive.

If you're using a high density disk, the dialog box won't give you a choice of sizes; it just has buttons labeled Eject and Initialize. There's only one way to initialize a high density disk, and the end result is a disk with a capacity of 1440K.

- ☐ **Click on Two-Sided or Initialize.**
- ☐ **When you see a box asking you to name the disk, type a new name.**

We'll use the name *Working Disk* in this example. But if you're working in a public lab, you should use your name and other ID information (for example, *Benjamin CS101 disk 1*) so you and your disk can be reunited if you are accidentally separated.

> You can always rename a disk icon (or any other icon on the Macintosh desktop) by clicking on the icon or its name and typing a new name.

Copying Files

- ☐ **Double-click on the new Working Disk icon.**

You should see an empty window, ready to fill with the stacks we'll be using in this session. (Once again, these instructions assume that files are arranged in folders as in the original HyperCard package. If your disks don't match these descriptions, you'll have to do a little improvising. If you're using your own personal hard disk, you'll probably want to use the files on the hard disk rather than the ones you put on this disk. If that's the case, think of this as your backup disk.)

- ☐ **Locate the Area Codes stack (probably in a folder called More Stacks) on your hard disk, file server, or HyperCard and Stacks disk. (If you're working from floppy disks, you may need to eject a disk in order to insert your HyperCard and Stacks disk.)**
- ☐ **If necessary, resize or move windows until you can see the Area Codes icon and the Working Disk window.**
- ☐ **Drag the Area Codes icon into the Working Disk window.**

You just told the Macintosh to make a copy of Area Codes on your disk.

Dragging an icon from one disk window to another copies that file onto the new disk *without disturbing the original*. If the icon is a folder, the folder and all of its contents are copied to the new disk. There are three situations where this kind of copying won't work:

1. The destination disk is locked or write-protected. (Solution: Unlock it.)
2. There's not enough space on the destination disk. (Solution: Delete or move some files from the disk, or use another disk.)
3. A file already exists in the destination window with the same name as one you're copying. A dialog box will ask you if you want to "Replace items with the same names with the selected items?" (Solution: Change the name of one of the files before making the transfer, or else say goodbye to one. To change a file name, just click on the icon and type a new name.)

When the copying process is completed, you should see two Area Codes stacks: the original and the new one on Working Disk. You can make changes to the new one whenever you like without changing the original.

If you want to copy several files from one window to another without copying all of them, you can do some selective selecting. In the same folder you should find stacks called Plots and Documents; you can copy both onto Working Disk at once.

- ☐ **Click on the icon called Plots.**

You can **shift-click** on another icon to select it without letting go of the currently selected icon.

- ☐ **Hold down the Shift key and click on the Documents Icon.**

Documents is highlighted, and so is Plots. Now you can move both of them at once.

- ☐ **Drag either one of the selected (highlighted) icons to the window of Working Disk.**

Since they're both selected, the other will follow. Now let's move the icons in the HyperCard Stacks folder.

- ☐ **Locate the folder called HyperCard Stacks on your hard disk, file server, or HyperCard and Stacks disk.**
- ☐ **Double-click on the folder to open its window.**

- **Select View by Icon and resize the window to so you can see icons for these stacks: Home, Address, DateBook, File Index, Phone, Quotations, and HyperCalc).**

  We want to copy these icons to Working Disk. We could drag the HyperCard Stacks folder icon to Working Disk, making an exact copy of the folder and everything in it. Or, since this window is open and active, we could use the **Select All** command to select everything in the window and then shift-click on any other icons that we don't want to include. (Try it, if you like; then click on white somewhere in the window to deselect them.) But for the sake of learning, let's use a different technique.

- **Use the pointer to drag a diagonal line from a point in the window above and to the left of the group of icons to a point below and to the right of them. (Start above and to the side of the icons; if you start dragging on one of the icons, you'll just drag the icon to a new location).**

  A dotted rectangle will form between the two points, showing you the area of the window you're selecting. When you release the mouse button, the icons inside that rectangle turn black, indicating that you've selected them all. The rectangle doesn't need to surround the icons completely to select them. But any icons that are even partially included in this rectangle are selected, whether you want them or not.

- **If your rectangle of selected icons includes any that are not in the list above, hold down the Shift key and click on each unwanted icon to deselect it.**

  You should have 7 icons selected now: Home, Address, DateBook, File Index, Phone, Quotations, and HyperCalc.

- **Drag any of the selected icons onto Working Disk; they will all be transfered.**

  Here's a selection summary:

Four Ways to Select Multiple Icons in an Active Window

- Choose Select All from the Edit menu (or type Command-A).
- Drag a rectangle around them.
- Shift-click on them in succession.
- Select a larger group and Shift-click on selected icons to deselect them.

Note: These techniques for selecting multiple icons *don't* work in HyperCard because HyperCard icons represent buttons.

Organizing with Folders

Time to get organized.

- ☐ **Click on the Working Disk window or icon to make it active.**
- ☐ **Select New Folder from the File menu (or type Command-N).**

A new folder should appear in the disk window.

- ☐ **Type "Session 2 Stacks" to rename that folder.**

If the name doesn't change, click on the icon or the name under it and try again.

- ☐ **Drag all of the stacks on that disk except Home into that folder's icon.**

The icons disappear, but they aren't really gone.

- ☐ **Double-click on the Session 2 Stacks folder icon.**

The window opens, revealing the icons that had disappeared.

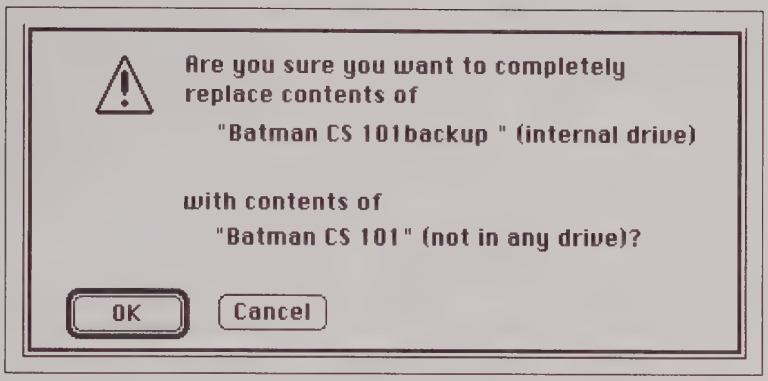
> When you drag an icon into the window or icon of a folder on the *same* disk, the original is **moved**—not copied—into that folder.

You can create folders and move your personal stacks into them to help prevent desktop clutter. If you use meaningful names and an intuitive organizational scheme, you'll have no trouble locating those stacks when you need them.

Backing Up a Disk

Before we get started with HyperCard, let's make a backup copy of Working Disk, if for no other reason than to learn how.

- ☐ **Select Eject from the File menu (Command-E) and insert your second blank disk into the vacated drive.**
- ☐ **Initialize that disk and give it a different name.**
- ☐ **Drag the icon for your Working Disk onto the icon for this newly initialized disk.**


Are you sure you want to completely replace contents of
"Batman CS 101backup " (internal drive)
with contents of
"Batman CS 101" (not in any drive)?
OK
Cancel

☐ **Click OK in the dialog box or press the Return key.**

> A button in a dialog box surrounded by a thick dark line represents the **default option**; pressing Return on the keyboard has the same effect as clicking that button.

All of the files on your Working Disk will now be copied onto the backup disk, making an exact **copy** of your Working Disk.

> When one disk icon is dragged on top of a second icon of the same type, the contents of the second disk are replaced with the contents of the first. The first remains unchanged.

Using the Trash

There are two stacks in your Session 2 Stacks folder that you won't really need; they're just taking up disk space. You can free that space by deleting the unwanted stacks.

☐ **Click on the Plots icon to select it. Then Shift-click on the Documents icon so both icons are selected.**

☐ **Drag either into the Trash icon; they'll both go.**

When your pointer reaches the trash can, the can will turn black, indicating that the selected items are now inside it. (Open the can to check if you like.)

If you're having second thoughts, you can drag either or both of them out of the Trash window and back to their original disk or folder window. (The stacks are still on the disk, even though the icons disappeared from the window. This feature can come in very handy—even the best of us occasionally have to dig through our own garbage.) Or, if you're ready to really dispose of them, you can move the pointer up to the Special menu and drag it down to **Empty Trash**. (Normally, it's not necessary to empty your own trash unless you're in a hurry to free some disk space. The machine will take care of it for you eventually.)

☐ **Drag the icon of your backup disk into the trash.**

You aren't trashing the disk; you're simply telling the computer you don't want to use it right now. The disk pops out and its image disappears from the desktop.

Customizing Your Work Space

The disk that remains is your working disk. If you followed instructions correctly, it should contain Home and a folder called Session 2 Stacks; the folder should contain Address, DateBook, File Index, Phone, Quotations, HyperCalc, and Area Codes stacks.

☐ **The last step depends on what kind of system you have. Choose one:**

- **If you're using a file server from a machine with two disk drives, you're ready to go.**
- **If you're using a hard disk in a public lab, you're ready to go.**
- **If you're using two HD disks in Superdrives, and you have HyperCard and the System Folder on one disk, you're ready to go. (You may want to drag Home to your system disk, too, if it will fit, to avoid having to answer the "Where is Home?" question each time you start HyperCard. If you do, delete it from your working disk.)**
- **If you're using your own personal hard disk system, all you need to do is make sure your hard disk contains those files. If they aren't already there, simply drag the two icons (Home and Session 2 Stacks) into your hard disk window.**
- **If you're using a file server from a machine with one disk drive, you can avoid unnecessary disk swapping by copying Home and the Session 2 Stacks folder onto your System disk. They probably won't fit unless you remove most of the files from the**

folder. (It doesn't hurt to try; the Mac will politely refuse if there's not enough room.) If they don't fit, try copying only Home, Address, Area Code, DateBook, and Phone; then create a new folder on your System disk called Session 2 Stacks and drag all but Home into that folder. If you do the projects at the end of the session, you can erase some of these files to make room for the files required for those projects.

- **If you're using a system with two disk drives, drag all of the stacks *except* Address, Area Code, DateBook, and Phone out of the Session 2 Stacks folder and into the Trash; then try to copy this folder and Home onto your system disk. If you can fit all of those on one disk, your other drive is free for the HyperCard disk. If that won't work, you might try creating a temporary disk containing HyperCard, Home, and Session 2 Stacks.**

Ready to go? Let's go back to HyperCard.

Home Base

Starting at Home

When you open HyperCard by double-clicking on the HyperCard icon, the first thing it does is look for Home. If it can't find it in the immediate neighborhood, it will probably ask you in a dialog box, "Where is Home?" To avoid having to navigate this dialog box (and to make sure HyperCard uses your personal copy of Home) you can double-click on Home, rather than HyperCard, right from the start.

☐ **Double-click on the icon of *your* copy of Home in your disk window.**

> Double-clicking on a stack icon on the desktop tells the Macintosh to do three things:
> - Launch HyperCard.
> - Locate the Home stack (even if that's not the stack that was double-clicked).
> - Locate and open the stack that was double-clicked.

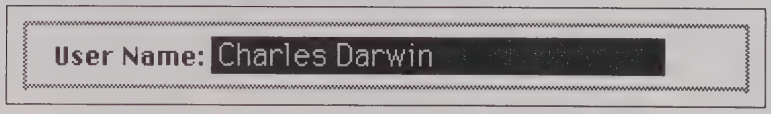
Changing the User Level and Name

We'll start by resetting the user level so we can do some typing.

- ☐ **Type Command-4 (Go Last).**
- ☐ **Click on the Typing button.**

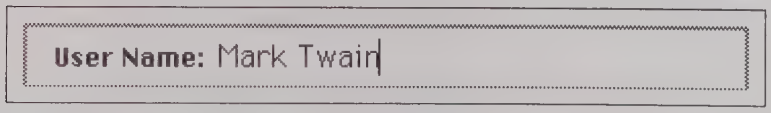
Typing user level allows you to type new information into HyperCard stacks and change existing information, while still maintaining your rights and privileges as a browser. It won't let you accidentally modify pictures or buttons. To test your new typing privileges, let's change the user name.

- ☐ **Select the name by dragging the mouse from the first character in the first name to the last character in the last name.**



It should appear highlighted in black.

- ☐ **Type your name.**



The selected text will vanish, only to be replaced by your typing. A flashing cursor will leave a trail of characters, just as you type them. If you make any typing errors, use the **Delete (Backspace) key** to erase the offending characters.

Changing the Home Memory

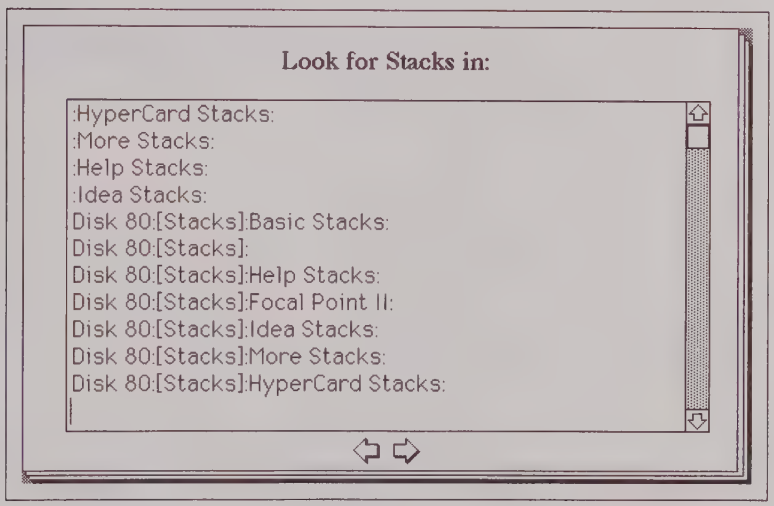
If you'll be using the disk you created as a working disk, you need to take care of one more piece of business to make sure your Home stack doesn't outsmart you later.

> HyperCard uses the Home stack to keep track of where you store your other stacks. When you click on a button to open a stack, HyperCard looks for that stack in the folders where it has found other stacks in the past. If it can't find it in those places, HyperCard asks you to locate the stack. When you find the stack in a folder, HyperCard adds that folder to its list of stack storage places.

Here's the problem: The Home stack that you copied may remember where the *originals* of all your copied files are stored, so it will go to those originals, rather than your copies, when you click on their icons. You can prevent that by erasing that part of your Home stack's memory. Here's how.

- ☐ **Type Command-3 twice to go past the first card to the second card in the Home stack.**

You'll probably see a cryptic list of disk and folder names like this:



- ☐ **Drag the mouse from the beginning of the first item in the list to the end of the last item in the list.**

If the list is too long to fit on the screen, it will probably scroll as you drag. All of the text you selected should be highlighted in black.

- ☐ **Press the Delete (Backspace) key.**

Everything that was highlighted in black should be gone, so HyperCard won't automatically look in those folders for stacks. You're about to teach it where to look for the files instead.

- ☐ **Type Command-1 to go to the first card in the stack: the Home card.**

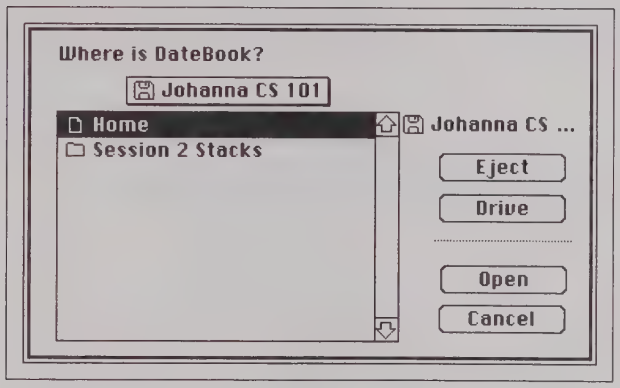
Locating the HyperCard Calendar

HyperCard is often used as a personal information management tool,

combining the functions of a date book, an address book, a notepad, and a file cabinet.

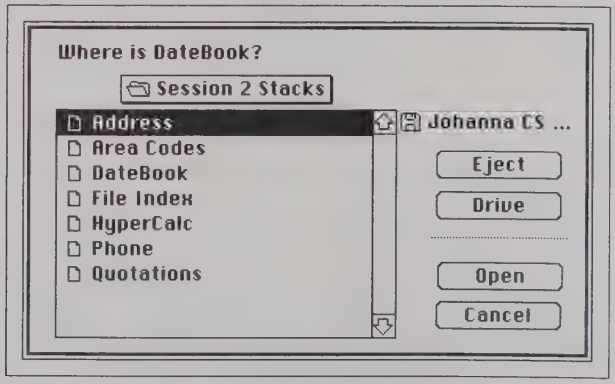
☐ **Click on the Calendar icon on the Home card.**

If you followed the instructions above, you should see this dialog box:



You put the DateBook stack, along with all the others, in the folder called Session 2 Stacks.

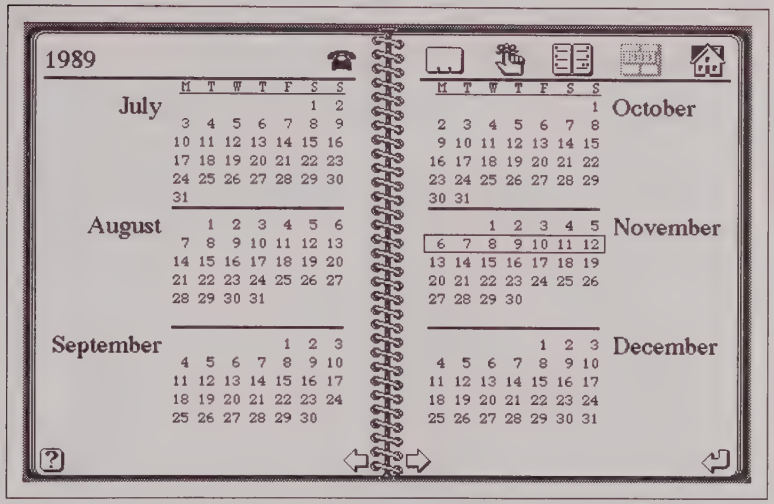
☐ **Double-click on that folder icon, and you'll see this:**



☐ **Double-click on DateBook.**

(You won't need to repeat this every time you open a stack, because your Home stack will remember where it found this stack.)

Once it finds the stack, HyperCard shows you a desktop calendar in a spiral book. The actual name of this stack is DateBook, and you're not looking at the first card of the stack; the smarter-than-average Calendar button took you to the card showing the current six-month period.



When you're working with HyperCard, the name and boundaries of the current stack aren't always apparent. Going to a card with a different background doesn't necessarily mean that you've gone to a new stack.

There are plenty of buttons on this card, including familiar buttons for going Home, going forward in the stack, going backward in the stack, and going back where we came from.

- **Click on the right-arrow on this card to go forward in time (repeatedly, if necessary) until the calendar runs out.**

You're still in the same stack, but you're probably looking at a weekly appointment calendar now. You *could* continue to step forward through the appointment book until you reach today's date. But that's the kind of sequential navigation that's only necessary with linear media like books and paper calendars; there's a better way in HyperCard.

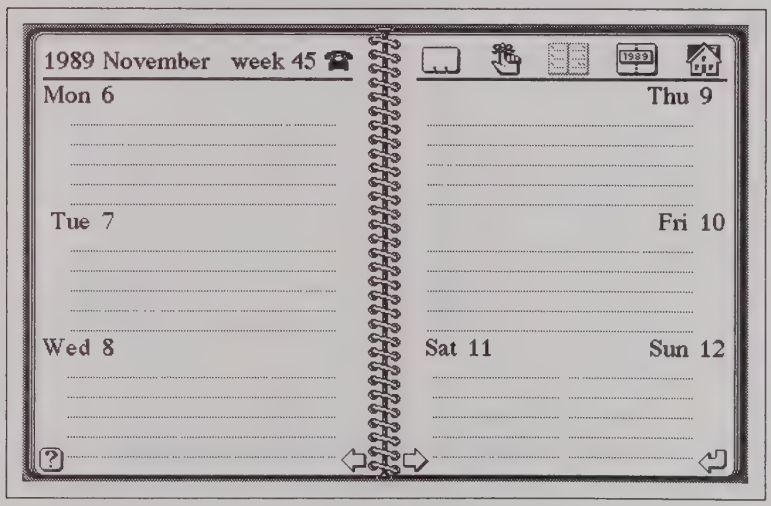
- **Press the Esc or tilde (~) key to go back to the current six-month calendar.**

- **Click on today's date on the calendar.**

  You should be transported to this week's appointment calendar card if that card exists.

- **If you see a dialog box that asks you if you want to "Make another six months of new cards?", click OK. When you're asked, tell it to start from today's date.**

  HyperCard will generate cards to bring the stack up to date.



HyperCard's Text Fields

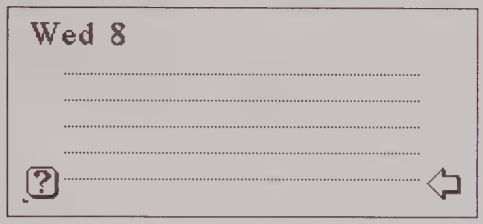
Entering information in the appointment calendar is straightforward; it involves the same kind of text-editing techniques that you use in Macintosh word processors, outliners, and databases.

Entering Text into a Field

Notice that each date in DateBook has a lined area for recording appointments, reminders, and other messages. Those lined areas are called *fields*, and they're similar to the fields of a conventional database.

- **Move the pointer over the lines for today's date.**

  Notice how the pointer turns from a pointing hand into an **I-beam** as you move over the field.

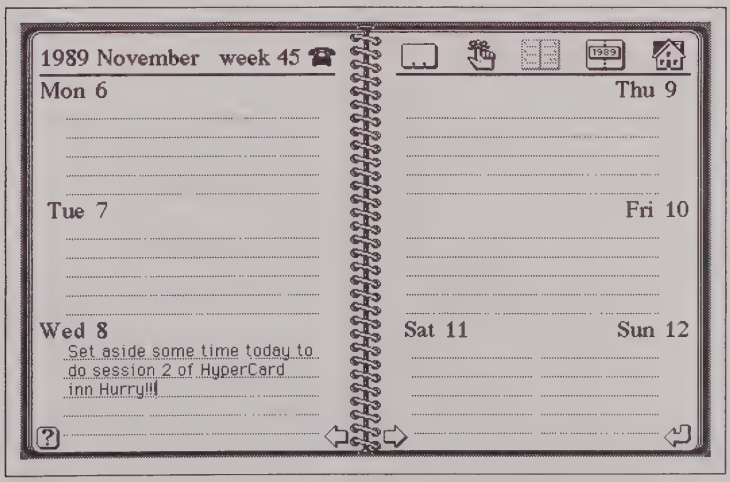


- **Position the I-beam at the beginning of the first line of the field for today's date and click once.**

As you move the I-beam away, you leave behind a flashing **insertion bar**. That's the spot where you'll type your first note, complete with a few errors to correct:

- **Type "Set aside some time today to do session 2 of HyperCard inn Hurry!!!"**

Don't press the Return key when you reach the end of a line.



Like most word processors, HyperCard fields automatically wrap text around to the next line when you reach the end. The only time you need to press Return is when you want to force something to go to the next line before the current line is full.

If you're familiar with Macintosh-style text editing (inserting, deleting, and changing), you may want to skip to "Using the Clipboard."

Editing Text

The three exclamation points are a bit much, so use the Delete (Backspace) key to delete two of them. This is the simplest method of **editing text**. You can backspace as far as you like to correct errors, but it's usually faster and easier to reposition the insertion bar.

- ☐ **Place the I-beam immediately to the right of "inn" and click to reposition the insertion bar.**

Don't confuse the I-beam with the insertion bar. The I-beam is for positioning the insertion bar. But until you click, the insertion bar is back where you left it.

- ☐ **Press Delete (Backspace) to turn "inn" into "in".**

When you **delete text**, everything to the right shifts left to fill the void. **Inserting text** is just as easy.

- ☐ **Type a space followed by "a".**

By first deleting text and then typing something new, you can **replace** any text in the field. But it's often easier to **select** the offending text first and then type the replacement.

Drag the pointer through the first *s* in *session*.

The *s* is highlighted, indicating that it is selected. Anything you type now will replace it.

- ☐ **Type "S".**

Selecting a whole word is easy; you don't need to drag exactly from the beginning to the end of the word.

- ☐ **Double-click on "do" to select it.**
- ☐ **Type "complete".**

You can combine the double-click and the drag to select multiple words.

- ☐ **Double-click on "some", holding the button down on the second click, and drag through "time", and type "two hours".**

Four Ways to Select Text

- Drag the pointer from the beginning to the end (or from the end to the beginning) of the text to be selected. If the text crosses several lines, move directly down to the end of the selection without crossing all of the characters on each line.
- Double-click on a word to select only that word.
- **Double-click-drag** the pointer to some other word; both (whole) words will be selected, as well as everything in between.
- Click on one end of the area to be selected and Shift-click on the other end; this has the same effect as dragging between those two points; as you'll see later, it's especially handy when part of the text to be selected is scrolled off of the screen.

Using the Clipboard

You can do more than delete or change selected text; you can copy it or move it using the Macintosh **Clipboard**. The Clipboard is an invisible portion of the Macintosh memory that can be used to store text, pictures, buttons, fields, and other forms of information so that they can be easily moved, duplicated, or reused. Macintosh veterans should be familiar with the basic Clipboard commands:

Macintosh Clipboard Commands

Copy (**Command-C**) copies the selected text, picture, or object to the Clipboard, leaving the original unchanged.

Cut (**Command-X**) places the selected text, picture, or object in the Clipboard, deleting the original.

Paste (**Command-V**) takes a copy of the contents of the Clipboard and places it in the current position. Paste does *not* empty the Clipboard.

Note: The Clipboard can hold only one piece of information at a time, so anything on the Clipboard is replaced when you issue a Copy or Cut command.

Another Note: In HyperCard, the Clipboard commands in the Edit menu change, depending on what's selected. For example, if text is selected, the Copy command is Copy Text.

Let's use the Clipboard to clone today's reminder.

- ☐ **Select the entire sentence by dragging straight from the beginning of the first word to the end of the last word. (You do *not* need to drag though every word.)**
- ☐ **Select Copy Text (Command-C) from the Edit menu.**

Your screen doesn't change, but a copy of the selected text is hiding in the Clipboard. To retrieve something from the Clipboard, use the Paste command.

- ☐ **Position the I-beam over the field for next Friday's date and click to place the insertion bar there.**
- ☐ **Select Paste Text from the Edit menu.**

A copy of the reminder appears in the new date.
Since the Clipboard still has the message, we can fill in the next day, too.

- ☐ **Position the insertion bar on the Saturday/Sunday field (remember to click), and type Command-V, the keyboard shortcut for Paste.**

Think of the *V* as an insertion caret to remember this shortcut.

- ☐ **Change the 2 in the Friday message to 3, so it reads "Session 3."**
- ☐ **Change "Session 2" to "Session 4" in the Saturday message.**

If you decide that you want to go to the beach on Saturday, you can easily move your Session 4 commitment to Monday.

- ☐ **Select the complete Saturday message.**
- ☐ **Press Delete (Backspace).**

Wait! If you clear out Saturday's message that way, you'll have to retype it again in Monday's slot. But we can get it back.

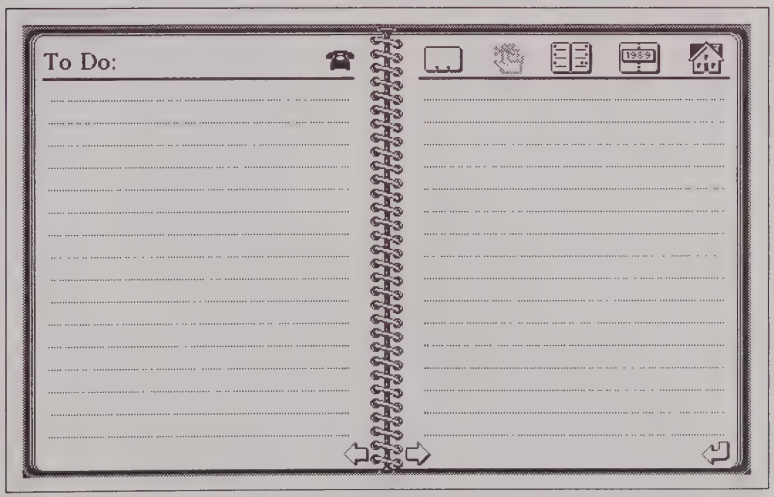
- ☐ **Select Undo from the Edit menu to bring back the deleted text.**
- ☐ **Select the complete message again and type Command-X (Cut Text).**
- ☐ **Turn the page with the right-arrow button and position the insertion point on Monday's field. Type Command-V (Paste Text).**

The **Undo** command (**Command-Z**) will *usually* restore your work as it was before you executed your last command or did your last round of typing. If you're human, you should probably memorize the keyboard shortcut, Command-Z, right away, so you have it at your fingertips.

But Undo can't solve all your problems; it can only take you back one step, so it won't help if you've done anything since your last error. And it's not always clear what "back one step" means. For example, HyperCard doesn't always recognize that you've completed a step when you're typing information in a field until you start working outside of that field.

You might want to keep a copy of this reminder on your To Do List.

- **Click on the forget-me-knot reminder icon.**



You turned the page in your DateBook to the To Do list.

- **Type Command-V to paste the reminder here.**

You can use the To Do list to keep a list of items in order of priority.

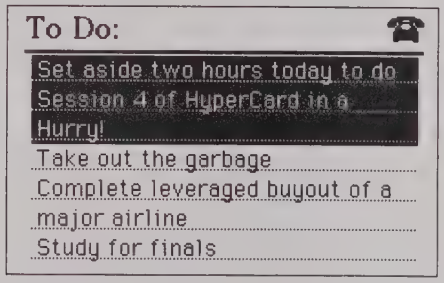
- **Position the insertion bar at the end of the Session 4 reminder and press Return.**

- [ ] **Add three or four items to the To Do list, pressing Return after each one to start the next one on a new line.**

> The Return key types an invisible **Return character** that tells the computer to move subsequent text to the next line. This character can be deleted or edited just like other characters.

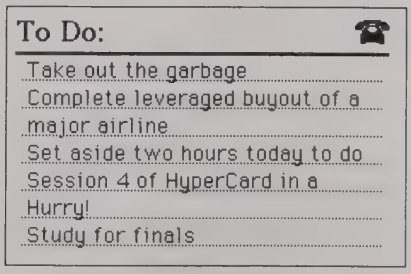
Let's use the Clipboard once more to rearrange the order of the items in your list so you can see how Return characters can be edited.

- [ ] **Position the pointer at the beginning of the Session 4 reminder and drag straight down to the beginning of the first line of the next item in your list.**



The Session 4 reminder is now selected, including the Return character at the end of it.

- [ ] **Type Command-X to cut the selected text.**
- [ ] **Move the insertion point to the beginning of any remaining item in your To Do list and type Command-V (Paste).**



You can rearrange priorities all you want like this.

If you've ever lost work when something went wrong in the middle of a computing session, you may be thinking that it's time to save what we've done so far. If you were working with a normal Macintosh application, your instincts would be right. But HyperCard is different.

> HyperCard regularly saves changes to a stack *automatically*, but it doesn't automatically save backups. If you're about to do something that you think might cause irreparable damage, or if you want to make a backup of the current stack on another disk, you can use the **Save a Copy...** command.

Basic Database Operations with HyperCard

HyperCard can be used for keeping track of people, too.

Browsing through the Address Stack

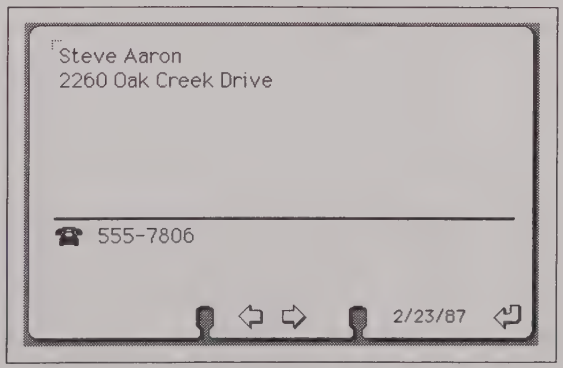
- ☐ **Click on the icon that looks like a Rolodex card.**

  This is your digital little black book.

- ☐ **To flip through the pages, click on the flying cards icon.**

  Too fast?

- ☐ **Use the right-arrow to browse through a few of the cards in the stack.**



Each card has three fields: one for name and address, one for phone number, and one for the date of the most recent change to the card. These fields don't have visible lines, but they can be edited in the usual way.

☐ **Try changing one of the addresses in the stack. When you're done, click outside the field somewhere.**

Notice that the date field changed by itself to today's date. This stack will keep that field current automatically for you.

Adding and Deleting Cards

For a Rolodex to be useful, you have to be able to add and delete cards.

☐ **Select New Card from the Edit menu (or type Command-N).**

> **Adding Cards**
>
> **New Card (Command-N)** creates a copy of the current card and locates it immediately after that card. If the current card contains text in fields, the text is *not* copied into the new card.

The insertion point is automatically positioned at the beginning of the name/address field.

☐ **Type your first and last names and press Return.**

> In name-and-address stacks like this one, it's important that the names be entered in the same order (for example, Firstname Lastname) on every card if you want to be able to alphabetize them later.

☐ **Type your address, pressing Return at the end of each line except the last. After you've completed the last line, press the Tab key.**

> The **Tab key** advances the insertion bar automatically to the next field in the card. If the field contains text, the text is highlighted, so anything you type will replace it. Tabbing from the last field takes you to the first field. **Shift-Tab** moves you *backward* through the fields of a card.

In this case, the next field is the phone field.

- ☐ **Type your area code and phone number.**

Let's add one more card, using the keyboard instead of the menu.

- ☐ **Type Command-N (New Card) and fill the card with another name and address.**

Deleting cards is almost as easy as adding them.

- ☐ **Select Delete Card from the Edit menu. (or Type Command-Delete).**

> **Deleting Cards**
>
> **Delete Card (Command-Delete)** is a potentially dangerous command. Undo will *not* restore a deleted card. If you have any doubt about whether a card should be deleted, you can use the command Cut Card instead. Cut Card has the same effect as Delete Card except that it places a copy in the Clipboard (replacing whatever was there before) so you can Paste it elsewhere in this stack or another stack.

Using Cut Card and Paste Card might be a good way to rearrange the cards in alphabetical order, for example. It might be, but it isn't, because there's a much more convenient way: the Sort button.

Sorting a Stack

- ☐ **Click on the icon that shows a hand rearranging cards—the Sort button.**

You'll see a dialog box that asks you if you want to...

- ☐ **Click on First Name.**

Almost instantly, the stack of names is rearranged in order by first name.

- ☐ **Browse through the cards until you find the one with your name.**
- ☐ **Click on the Sort button again and sort by Last Name.**

Now that they're in order, it should be easy to find your own card in the stack by flipping forward to your part of the alphabet again. But there's a better way.

Using the Find Command

- ☐ **Select Find... from the Go menu (or type Command-F).**

The long box at the bottom of the screen, called the **message box**, now has an incomplete message in it; the Find command, waiting for you to tell it what to find. Later you'll learn how to type all kinds of HyperCard commands, but for now all you need to do is fill in the space between the quotation marks.

- ☐ **Type your last name, followed by a Return.**

Faster than you could type it, HyperCard will find the first card in the stack with that name. If it finds a card for somebody else with the same name, just press Return again to repeat the command.

You don't need to type a complete word into the Find command; you just need to type enough characters so that the term you're searching for can be identified uniquely. Suppose you want to find *Florence*.

- ☐ **Double click on your name in the message box to select it.**
- ☐ **Type "Flo" and press Return.**

Flo replaces your name in the Find command. Hypercard may or may not find Florence, depending on where you started your search. It might also find Kenneth, who lives on Flood Circle, or anybody who lives in Florida.

- ☐ **Press Return again several times, watching what happens on the screen as you do. Stop on Florence.**

Find Command Tips

When you use the Find command (Command-F), each time you press Return you're taken to the next card in the stack with characters at the beginning of any word in any field that match the pattern you typed between the quotation marks.

Two cautionary notes:

- You get what you asked for, even if it's not what you wanted.
- Conversely, you don't get what you don't ask for, so make sure you ask for exactly what you want. (The one exception: Case is irrelevant).

Example: If you say Find "cath", you might find "Cathy" or "cathedral" or "cathode" but not "Kathy".

The moral: Choose your search patterns carefully!

We can easily check to see if anybody else in the address book lives in your state, and learn another finger-saving shortcut at the same time.

- ☐ **Use the Find command (Command-F) to find your last name again.**
- ☐ **Type Command-F to highlight your name in the message box.**
- ☐ **Hold down the Command key while you click on your state's name.**

Notice what happened: The pointer kept its hand shape and your state's name was typed automatically into the message box, replacing the selected text.

Command-Click and Command-Drag

When you hold down the Command key while dragging or clicking on field text, that text is typed automatically into the message box. If text is highlighted in the message box, that text is replaced. (You can remember this by thinking about it as if you were saying, "Turn the text I'm selecting into (part of) a command."

- ☐ **Press Return to locate other cards with your state name. Keep going until you return to your card.**

Using the Phone Dialer

Once you locate the card you're looking for, what can you do with it?

- ☐ **Select the phone number by dragging through it.**
- ☐ **Click on the Phone icon.**

If you happen to have your phone properly connected to the computer's sound jack, or if you hold the phone close enough to the computer's sound source, those beeps will actually dial the selected number for you. The dialing is done by the phone stack; you're temporarily transported there when you push the Phone button, but you're automatically returned to where you started when the dialing is done. You can customize the dialer by going back to the phone dialer card.

- ☐ **Press the Go Back key (Esc or ~).**

If this card doesn't make complete sense to you,

- ☐ **Click on the Question Mark icon.**

Pop-up instructions will tell you just about everything you need to know about the dialer.

> A **Question Mark button** generally offers help with the current card.

- ☐ **Click on the instructions to make them disappear.**
- ☐ **Click on the Area Codes button.**

This is a reference stack, designed to be read but not changed. The fields are locked to prevent erasure, so the pointer doesn't change to an I-beam when you move it over them.

- ☐ **Type Command-F (Find).**

The Find command appears in the message box with your last search pattern—your state's name—highlighted.

- ☐ **Press Return, repeatedly if necessary, until your own area code is highlighted.**

Each card you see should have a city or state that matches the pattern. Let's use the message box instead of the menu bar to go home.

- ☐ **Select all of the text in the message box.**

□ **Type "Go Home" and press Return.**

You'll learn more about how to use the message box to communicate with HyperCard in later sessions.

Using the Printer

It's certainly handy to have HyperCard instantly locate appointments, find names, and dial phone numbers . . . as long as you're sitting in front of the computer. But if you use HyperCard to store names, phone numbers, appointments, and other essential information, you need to have a way of making a **hard copy** so you can take the information with you when your computer stays home.

Printing from HyperCard

HyperCard has three Print commands for making hard copies of what you see on the screen:

- **Print Card** prints the current card image — no more, no less.
- **Print Stack** prints all of the cards in the stack, with options for printing them full sized or reduced so more can fit on a page.
- **Print Report** prints a database-style **report** showing any or all of the information stored in text fields in the stack. This command allows you to control which fields will be printed, how they will be arranged in the report, and what information will appear in the header at the top of each page.

Each of these commands is explained in detail in the HyperCard Help stack. Before printing anything, you might need to use the **Chooser** desk accessory and the **Page Setup** command from the file menu to select a printer and control the layout of the page.

If your computer is connected to a printer, you might want to experiment with these commands.

Covering Our Tracks

We could spend more time exploring HyperCard's information management tools. But you've seen the basics now, and the exercises and projects will give you a chance to apply the principles you've learned in a variety of interesting ways. For now, let's review a few tricks as we put things back more or less the way we found them. (This is essential if you're using shared files, and recommended even if you aren't.)

- ☐ **Locate your name in the Address stack and select Delete Card from the Edit menu.**
- ☐ **Delete everything you typed on the To Do page.**
- ☐ **Delete all four of your DateBook entries.**
- ☐ **Select Quit HyperCard from the File menu.**
- ☐ **Select Shut Down (or Restart) from the Special menu.**

Summary

You've accomplished quite a bit in this session. You created a personal working disk for your stacks, learning basic Macintosh disk and file operations in the process. You learned how to enter and edit text in HyperCard fields using HyperCard's DateBook and Address stacks. You learned how to add and delete cards, sort cards in alphabetical order, and find particular items using character string pattern matches.

You're a certified HyperCard user now, prepared to browse and type your way through all kinds of HyperCard stacks. If you have no intention of creating your own stacks, you can quit now. But be forewarned: The best is yet to come.

Key Words

backup copy
Chooser desk accessory
Clipboard
Command-click, Command-drag
Copy Card command (Command-C)
Copy Text command (Command-C)
copying a disk (Finder)
copying a file (Finder)
Cut Card command (Command-X)
Cut Text command (Command-X)
data file
database
default option
Delete (Backspace) key
Delete Card command
deleting text
double-click-drag
editing text
Empty Trash command (Finder)
field (database and HyperCard)
Find command (Command-F)
hard copy
I-beam
initializing (formatting) a disk
inserting text
insertion bar
K
locked (field)
locked or write-protected (disk)
master copy
message box

moving a file (Finder)
New Card command (Command-N)
New Folder command (Command-N, Finder)
Page Setup command
Paste Card command (Command-V)
Paste Text command (Command-V)
Phone dialer
Print Card command (Command-P)
Print Report command
Print Stack command
Question Mark button
record (database)
replacing text
report
Return character
Return key
Save a Copy... command
Select All command (Command-A, Finder)
selecting text
Shift-clicking
Shift-Tab
Sort button
sorting a stack
Tab key
Trash
Typing (user level)
Undo command (Command-Z)
working copy

Self-Testing Exercises

1. There's no Save command in HyperCard. Is it necessary to use the Save a Copy... command to save your work? Explain.
2. What is the standard HyperCard meaning for each of these icons?

   a.

   b.

3. Give three examples illustrating *different* reasons why the Find command might not take you to the sought-after card.
4. Why is it important to make sure all of the names in an address book stack are listed with the first and last names arranged in the same order?
5. You created a folder for Session 2 stacks. Where is the folder located? How is it useful to have the folder? In what ways is a Macintosh folder similar to and different from a paper folder in a file cabinet?
6. Explain the difference between deleting and cutting a card.
7. What is the keyboard command to do the following?

   a. Open the HyperCard Help stack

   b. Go to the First card in the stack

c. Go Home

d. Go Back

e. Undo

Projects

Note: Some of these projects involve modifying standard HyperCard stacks. If you're working in a public lab with shared files, use the personal working copies of these files that you created at the beginning of this session. In order to make sure you open your copy, rather than te public copy, use the Open Stack... command to locate each file on your disk.

1. Open your copy of the Quotations stack. Add at least three new quotations to the end of the stack. Try them out to make sure they work like the others.

2. Open your copy of the Plots stack. This stack creates bar, coin, and pie graphs for any series of numbers entered into the scrolling field on the right side of the card. Use this stack to draw charts to represent the values for your personal bank account balance over the past 6 months, your term-by-term grade point average, your monthly expense budget, or the ages of several of your friends. For each problem, print the card when the chart is complete, and write explanatory captions on the printout.

3. Open your copy of the HyperCalc stack. This is a simple spreadsheet-style application that automatically recalculates results each time you change a number or a formula. It's set up now to calculate gas costs. How much does it cost per year to drive each of these vehicles?

   a. a 12-miles-per-gallon GMC van driven 15,000 miles per year with gas priced at $1.25 per gallon

   b. a 29-mpg Toyota Tercel driven 12,000 miles per year with gas priced at $1.10 per gallon

   c. your own vehicle at today's gas price

4. Open the File Index stack. Copy the word *sprocket* from the instructions on the first card and Paste it into a Find command to search for the word *sprocket*. Explore this stack to get a feel for how HyperCard can be used as a free-form note collector. Add a few cards with notes, quotes, and so on.

5. Open the Address stack and Save a Copy.... Delete all the cards from your copy and build a new Address stack containing your friends and relatives. Sort the stack in alphabetical order by last name.

6. There are thousands of HyperCard stacks in existence. If you have access to a library of HyperCard stacks (there may be one on your lab file server), locate and explore several stacks in your area of interest.

SESSION 3

Objectives

By the end of this session you should be able to

- Create and name a new HyperCard stack and save a copy of that stack
- Use the Polygon tool, the Rectangle tool, the Rounded Rectangle tool, the Straight Line tool, the Pencil tool, the Paint Bucket, and the Patterns palette to create a drawing
- Use the Paint Text tool to add different styles of text to a stack
- Use the Eraser, the Undo command, and other editing tools to edit the graphics in the stack
- Create a button with a labeled icon and link it to a new card in the stack
- Copy a portion of a picture and paste it on another card
- Copy a button and paste it onto another card

3

STACKS FROM SCRATCH:

BUILDING WITH BUTTONS

The Problem

Your local museum is preparing an exhibit on the seven wonders of the ancient world. They've asked you to prepare an interactive display on Egypt's Great Pyramid that will (1) show a cross section of the pyramid, revealing passages and rooms; (2) provide close-up views of the King's Chamber and other rooms; and (3) be interactive and self-explanatory, so anyone can explore it. You don't have the time or money to create physical models, and you don't have the wall space to hang drawings or photos. Perhaps HyperCard can help.

Introduction

So far we've focused on using HyperCard as a tool for storing and accessing information. The projects in Sessions 1 and 2 highlighted HyperCard's value as a reference tool and a personal information manager. But all of the examples in those sessions were built around HyperCard stacks that were created by somebody else to meet the needs of typical computer users. If we think of computer software as an extension of the human mind, then canned stacks like those are intended to extend the *average* computer user's mind. But the concept of "average" often has little to do with reality; a person with one foot in ice and the other in fire is not, on the average, comfortable. Programs written for average people aren't written for real people.

Fortunately, HyperCard makes it easy to customize existing stacks or create new stacks that can meet the exact needs of individual users. In fact, as you'll see in this session, it's almost as easy to create your own stacks as it is to browse through stacks created by others. HyperCard does such a good job of breaking down the distinction between computer *user* and computer *programmer* that you might not even notice when you cross that line in this session. Programming has never been easier.

Birth of a Stack

Resetting the User Level

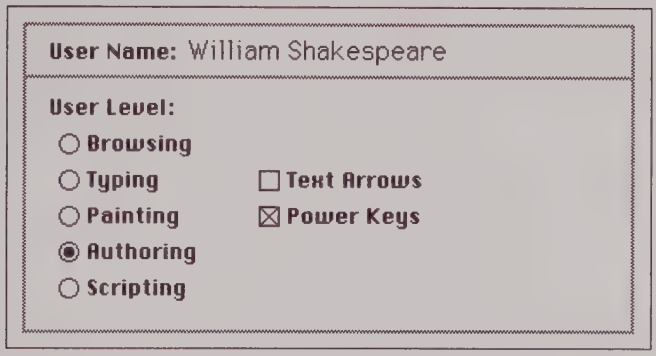
HyperCard's browsing and typing user levels are designed to protect stacks from being accidentally modified by untrained users. But starting this session, you'll need to set your user level to level 4, authoring. (Level 3 is **painting**; we're skipping that level because it doesn't allow you to create new buttons, cards, or stacks.)

- ☐ **Turn on your computer. If you're working with a hard disk, wait for the computer to boot from the hard disk and insert a disk containing your personal copy of Home and enough blank space to hold your stack creation (60K is plenty for this project). If you're working from a server or a disk system, insert your personal system/Home disk(s).**

We won't be modifying any of the standard HyperCard stacks for the rest of the book, so you won't need personal copies of those files in the remaining sessions.

- ☐ **Open the Home stack.**
- ☐ **Click the left-arrow button to go to the last card in the stack.**

- ☐ **Click on the Authoring radio button.**

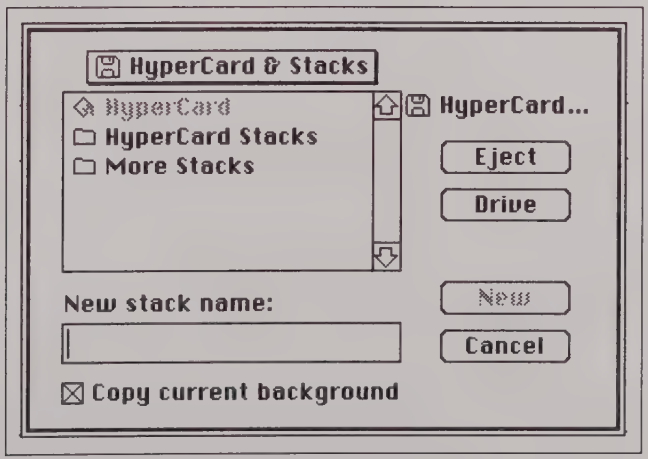


The **authoring** user level gives you access to all of HyperCard's tools except scripts, which will be introduced in Session 6. The **Power Keys** check box allows you to use all kinds of one-key commands rather than having to go to the menu bar while you're painting. We won't use them in this introductory session, but they're summarized inside the back cover so you can learn them when you're ready.

Creating a New Stack

Select **New Stack** from the File menu.

(If the New Stack command is dimmed in the menu, double check to make sure your user level has been set to authoring.) You'll see this:



Make sure the disk shown above the Eject button is the appropriate disk for saving the stack (probably your working disk or a blank disk). If it's not, use the Drive and/or Eject buttons in the dialog box to locate the disk so that it appears.

(If you're working on a file server and you have your system disk in the only available drive, you can save it on that disk to avoid swapping. Later, you can copy it to another disk to make room for your next stack.)

☐ **Type "Pyramid".**

This is the name you're assigning to your about-to-be-created stack.

> **Naming a Stack**
>
> You can choose any name for a stack, as long as
>
> - it is less than 32 characters long and contains no colons (:)
> - it is not the same as the names of any other files in the same folder on the disk
>
> It's usually a good idea to choose a name that describes the stack contents, so you don't need to actually open the stack to remember what it does.

☐ **Click on the check box to turn off the option labeled Copy current background.**

We're starting off with a plain white background with no buttons, which is what you get if you don't copy the background from the current stack. You'll learn more about how backgrounds work in the next session.

☐ **Click New.**

Except for the menu bar, the screen (or HyperCard window on a large-screen machine) will turn white. You're looking at the first (and only) card in your new stack. It's totally empty now, but it's still saved on disk. HyperCard will automatically save this stack on this disk regularly as you work on your stack.

Stack Construction Toolbox

Tearing Off the Tools Menu

You'll be heavily using the **Tools menu** for a while, so let's put it where it's more accessible.

☐ **Drag your mouse from the title of the Tools menu down past the bottom of that menu. Drag the dotted image of that menu to the right side of the screen near the top.**

As you can see, Tools is a **tear off menu.** You can now select tools from the Tools menu (it's still there) or the **Tools palette** that you just created. The top three tools in the Tools menu/palette are **general tools** that allow you to choose between browsing, working with buttons, and working with fields; all of the remaining tools are **paint tools** specifically designed for painting. Notice that the Browse tool—the pointing hand—is highlighted. That means that it's the selected tool—the one that's currently being controlled by the mouse. When you select any of the paint tools, you tell the computer you want to paint on the screen or you want to modify something that's already painted on the screen.

> Painting in HyperCard is really just the process of telling the computer which **pixels** (picture elements or dots) in the card should be black and which should be white. Those are the only two choices; shades of gray are made of different black and white patterns. Technically, this is called **bit-mapped graphics**, because each pixel on the screen is controlled by one **bit**, or binary digit of the computer's memory. HyperCard has many sophisticated tools to make it easy to paint complex shapes, lines, words, and patterns, but all of these tools are doing the same thing: telling HyperCard to display some card pixels white and others black.

Drawing Lines

☐ **Click on the Pencil tool to select it.**

The Objects menu disappears, only to be replaced by three other menus. These menus are used for painting, so they aren't visible when you aren't using paint tools.

☐ **Move the pointer over the HyperCard card—the white screen below the menu bar (inside the window if you're using a large screen).**

Notice how it turns into a pencil.

☐ **Hold down the mouse button and drag the Pencil around on the screen.**

- ☐ **Try signing your name, remembering that you can pick up the mouse and move it if you run out of desk space. Then underline your signature.**

As you can see, it's easy to draw with the mouse, but not particularly easy to draw *well*. Straight lines are particularly difficult. Fortunately, there's another tool designed to make that easier.

- ☐ **Select the Straight Line tool.**

The pointer turns into cross hairs, ready to draw.

- ☐ **Drag the mouse with the button down between any two points in the window.**

Notice how a straight line stretches like a rubber band between those two points. Wherever you let go, that's where the line stays. Draw a few more lines if you like. When your screen gets hopelessly cluttered, you can erase anything you don't like.

Using the Eraser

- ☐ **Select the Eraser tool from the palette by clicking on it. (If you prefer, you can select the Eraser in the Tools *menu*.)**

Notice how the pointer turns into a square when you move it over the white card.

- ☐ **Drag the Eraser through part of your scribbles with the mouse button down.**

The parts you erase disappear without a trace. But erasing like this can be tedious if you want to clear the entire screen. Fortunately, there's a shortcut.

- ☐ **Double-click on the Eraser tool.**

> Double-clicking on the Eraser clears everything that's painted on the card. This is a handy—and potentially costly—trick. If you ever do it by mistake, remember Command-Z, the Undo command.

By this time, you're probably anxious to do some serious doodling. If so, take a few minutes to experiment. When you're done playing, double-click on the Eraser to clear the screen and we'll build a pyramid.

Drawing Polygons

- **Select the Polygon tool.**

This tool works like the Straight Line tool, except that it draws a series of connected lines. Each successive mouse click forms another angle and starts a new line; the lines stop forming when you click on the starting point or somewhere outside the window or when you double-click.

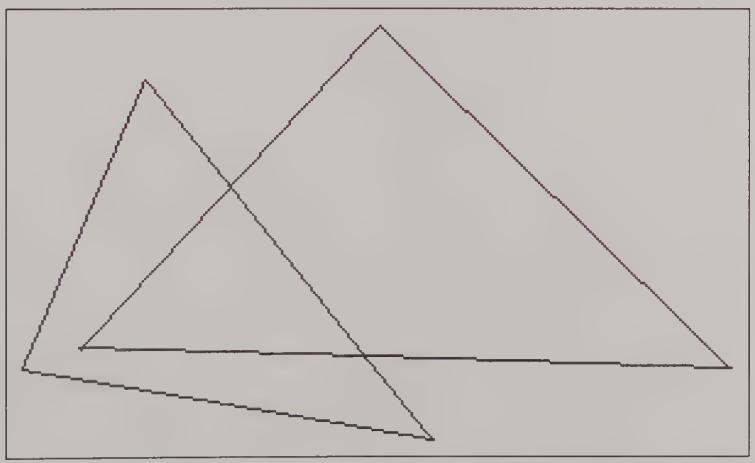
- **Draw a large triangle with the Polygon tool by clicking on three different points on the screen. Complete the triangle by clicking once more on the starting point.**

If your triangle doesn't look like a pyramid, never mind. You already know one way to remove it: You can double-click on the Eraser to clear the window. But Undo will work here, too:

- **Type Command-Z (or select Undo from the Edit menu).**

If you haven't clicked the mouse button anywhere else since you drew the last triangle, the triangle will vanish.

- **Draw two more overlapping triangles. Type Command-Z *three times* after you draw the second one. Watch how the picture changes each time you select Undo.**



Since Undo can't take you back more than one step, you'll never be able to remove that first triangle with it. That's why it's important to check your work after each step before doing any more typing, clicking, or dragging.

Graphics Insurance

When you're working with paint tools, it's easy to spoil a work in progress with an accidental slip of the mouse. You can avoid serious, irreversible damage by remembering and using these insurance commands:

The **Go Back** key (**Tilde** or **Esc**) backtracks one step when you have a paint tool selected. (To go back to the last card when you have a paint tool selected, type Command-Tilde or Command-Esc.)

Undo (Command-Z) backtracks exactly one step.

Keep (in the Options menu, or **Command-K**) tells HyperCard to save the current picture.

Revert (in the Options menu) tells HyperCard to go back to the last saved version of the picture.

When you're painting in HyperCard, your picture is saved to disk when you do any of the following:

- switch to another card
- switch to or from the background layer (You'll learn about back grounds in Session 4)
- select any of the three general tools at the top of the Tools palette
- select Keep (Command-K)

When you're painting, make a habit of selecting Keep every few minutes. If you make an error, try to erase it using Undo, Tilde, or Esc. If that doesn't take you back far enough, use Revert.

☐ **Double-click on the Eraser to clear the window.**

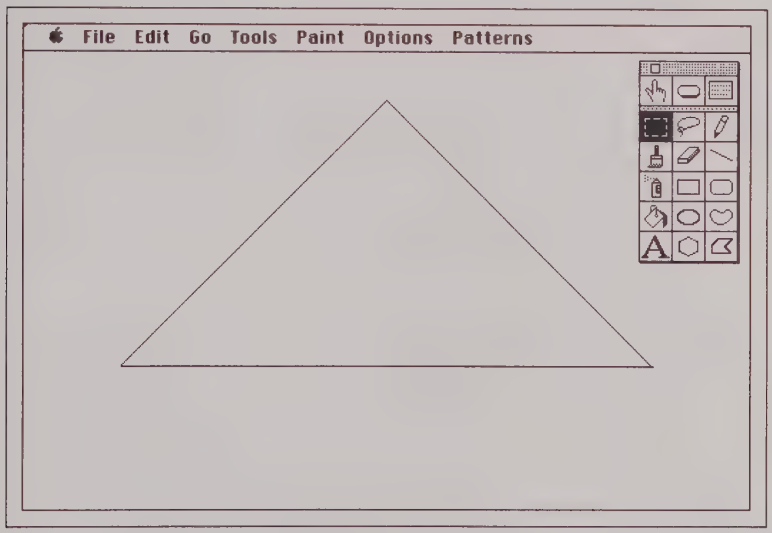
If you've been trying to make your triangles look like pyramids, you've probably noticed that it's not easy to make the two sides exactly the same length. Here's a trick that might help.

☐ **Hold down the Shift key while drawing the triangle with the Polygon tool.**

When you're using many of the paint tools, the Shift key serves as a **constraint key**, temporarily constraining the tool in some helpful way.

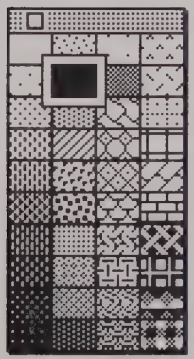
In this case, it constrains the Polygon tool so that every line is either horizontal or some multiple of fifteen degrees from horizontal. It's much easier to draw true horizontal lines and equal angles when the tool is

constrained this way; along with the freedom to position your line at any angle, you're giving up the responsibility of having to hit the vertical right on the mark. Even with this constraint, you'll probably have to try a few times to get the pyramid to come out right. If it doesn't work, just erase and try again. Your goal is to produce a triangle with two equal sides that fills about the top three-fourths of the window, like this:



(Your local Egyptologist might tell you that this pyramid doesn't exactly match the geometry of the Great Pyramid. Never mind; he just lacks perspective.)

Pouring Paint Patterns



Shape notwithstanding, the pyramid doesn't look authentic because it lacks texture. The Great Pyramid was built with gigantic granite blocks (enough blocks to stretch three-fourths of the way around the planet if laid end to end). We can simulate the granite blocks by painting our pyramid. Unlike real-world paint, HyperCard paint comes in patterns, rather than colors. A selection of these patterns is available in the **Patterns menu**, which can be torn off and made into a portable **Patterns palette**.

- **Drag the Patterns menu down from the menu bar and over to the right side of the screen so it's below the Tools palette.**

Notice that black paint is selected when you first open the palette.

☐ **Click to select the pattern that looks like bricks.**

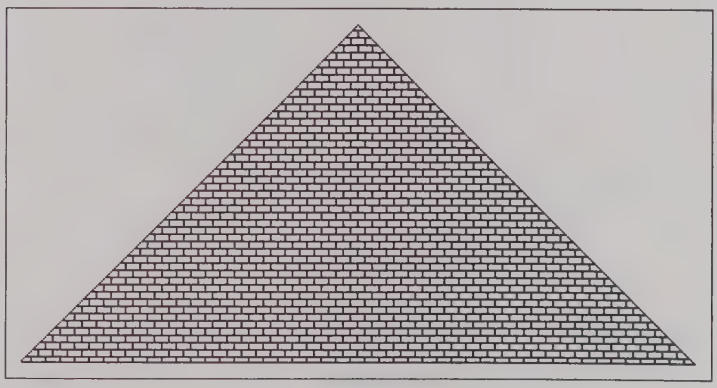
This is the paint pattern we'll use on the pyramid.

☐ **Select the Paint Bucket tool from the Tools palette. The pointer should turn into a paint bucket.**

The Paint Bucket tool allows you to pour paint into any white area that's completely enclosed by black (or black area that's enclosed by white).

☐ **Position the Paint Bucket so that the dripping paint is somewhere inside the triangle and press the button to pour the paint.**

Your triangle should fill with bricks. If the whole screen fills with bricks, your triangle has a leak.



The area to be filled with poured paint must be completely surrounded by black (or by white, if you're pouring into a black area). Even the tiniest pinhole in the boundary will allow paint to leak out to fill a larger area. If your paint ever leaks unexpectedly, select Undo *immediately*, locate and correct the leak with the Pencil (or redraw the shape), and pour the paint again.

We have a pyramid; let's add the desert.

☐ **Use the Pencil tool to draw a line from the left edge of the bottom of the pyramid to the left edge of the screen. Make sure that there are no gaps at either end of the line. Repeat on the right side of the pyra-**

mid. (If necessary, drag the Patterns palette out of the way using the bar at the top of the palette.)

These lines represent the desert's horizon in our landscape. For instant desert, just add sand.

- ☐ **Select the Paint Bucket tool and the pattern that looks like irregular grains of sand. Position the paint drip on the lower half of the screen to fill the desert with sand.**

Once again, if there's even the slightest pinhole in the horizon, you'll get more sand than you bargained for. If necessary, choose Undo, fill the leaks, and try again.

- ☐ **When your picture looks OK, type Command-K (Keep), just in case you need to revert to this picture later.**

Using Paint Text

Even though you may have no trouble identifying the Great Pyramid in your picture, others might need a hint. So let's add a title. You worked with text fields in Session 2, but for this job there's another kind of text that's more appropriate.

HyperCard's Two Kinds of Text

- **Paint text**, after it's typed, blends into the whole painting, becoming just another pattern of black and white dots on the screen. Consequently, it can't be edited later using the standard editing techniques. (Nor can it be found with the Find command.)
- **Field text** remains text forever. Unless a field is locked (see Session 5), you—or any user—can always edit it using the usual editing techniques.

Since we don't want casual users of our stack to mess with the title, we're better off using paint text.

- ☐ **Select the Paint Text tool (the letter A).**

The pointer turns into an **I-beam.** As you saw in the last session, the I-beam is used to position the cursor for typing.

- ☐ **Position the I-beam just below the bottom of the pyramid in the center and click.**

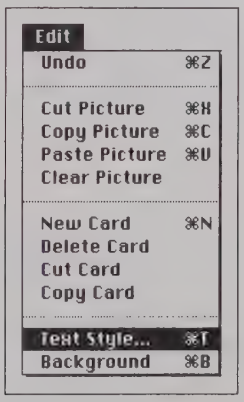
A flashing cursor appears, beckoning you to type.

- ☐ **Type "The Great Pyramid".**

The words are there, all right, but they lack style. We can fix that.

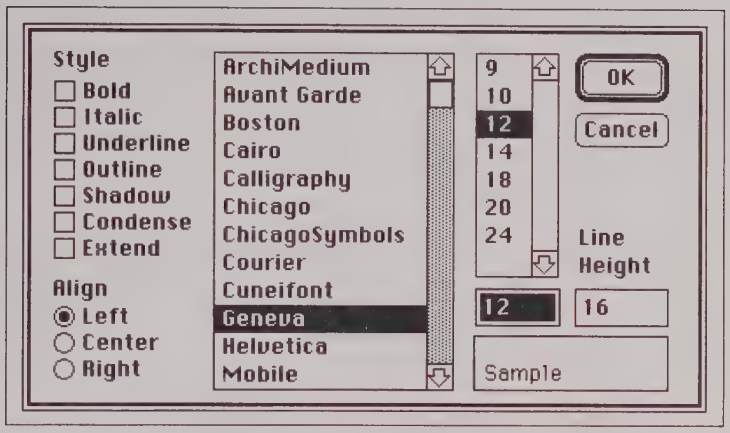
- ☐ **Double-click on the Paint Text tool.**

This is a shortcut to save time. The alternative is to select **Text Style...** from the Edit menu (or type **Command-T**).



You'll see a dialog box that looks something like the one at the top of the next page.

This crowded dialog box gives you control over the way the text looks, rather than what it says. If you haven't clicked anywhere since you typed your title, you can change it simply by clicking on different options. But first, a brief explanation of the available options is in order.



The Text Styles Dialog Box

The text **styles** listed on the left side—bold, underline, and so on—can be used to draw attention to text and add visual pizzazz to a screen. But beware: It's easy to overuse these styles and make your stack look, well, Mac-tacky.

The names in the center of the box represent fonts—basic typefaces—that are stored in your System file. Different fonts have different looks; compare this text with the text of the instruction on the next page that begins with "Experiment . . .". The font names that you see on your screen probably don't match those shown here; you may have different fonts stored in your System file. There are hundreds of fonts available for Macintosh users: Some come free with the System; others must be purchased; still others are freely available in the public domain. Consequently, it's rare for two computer systems to have the same set of fonts.

You can also control the **size** of the text by clicking on different sizes. (Sizes are measured in **points**; a point is a typographer's unit of measure.) Once again, the sizes that are available are determined by the fonts that are stored in your System. What's more, different fonts may have different available sizes. (In truth, you can type in sizes that aren't stored with the system for a particular font, but it's generally not a good idea. The computer will estimate what the font should look like based on the sizes that *are* available; the results are usually lumpy and hard to read.)

Finally, the dialog box lets you control **alignment** of the text. Normal text is aligned to the left, but you can also specify right alignment or automatic centering around the insertion point.

- **Experiment with the sample text in the lower right corner of the dialog box by choosing different combinations of fonts, styles and sizes. When you're done experimenting, click on Style Bold, Style Shadow, Style Extend, font Geneva, size 24 point (if it's visible; otherwise click on the largest available number), Align Center. Then click the OK button to apply all these changes to your title.**

The Great Pyramid

If the title didn't change, it's probably because you clicked somewhere else in the window before choosing Text Styles.... Remember, paint text "dries" as soon as you let go of it.

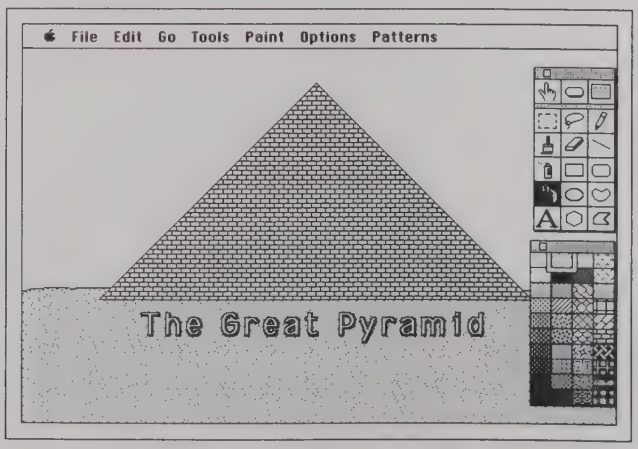
From an aesthetic point of view, these letters leave something to be desired. (Outline and shadow characters generally do.) But they'll serve our purposes because they stand out even when surrounded by sand.

Since the text cleared a path in the sand, we need to do some touch-up work with the Paint Bucket.

- **With the paint bucket and the sand pattern selected, position the paint drip over part of the area whited out by the text and click to pour.**
- ***Carefully* position the end of the paint drip inside the opening in the *P* and pour; fill the *d* the same way.**

If you make any mistakes, remember to select Undo before you click anywhere else.

If your screen looks like this, you've completed the artwork on your first card:



Creating a Button

- **Click on the Button tool in the top row of the Tools palette.**

The Paint, Options, and Patterns menus are replaced by the **Objects menu**, and the pointer turns into the familiar arrow.

- **Select New Button from the Objects menu.**

This is the generic HyperCard button. It's up to us to make it look and act the way we want it to. Let's start by making the button bigger.

- **Place the pointer at the corner of the button and drag it away from the button's center.**

If you found the right spot, the button should stretch as you drag. If it doesn't, reposition the mouse and try again.

- **Drag the corner until the button is about twice its original length.**

This button is inconveniently located right on top of our pyramid, but it's easy enough to move.

☐ **Place the pointer in the middle of the button and drag it down between the bottom of the card and the title.**

What does this button do?

☐ **Try clicking once on this button to see what it does.**

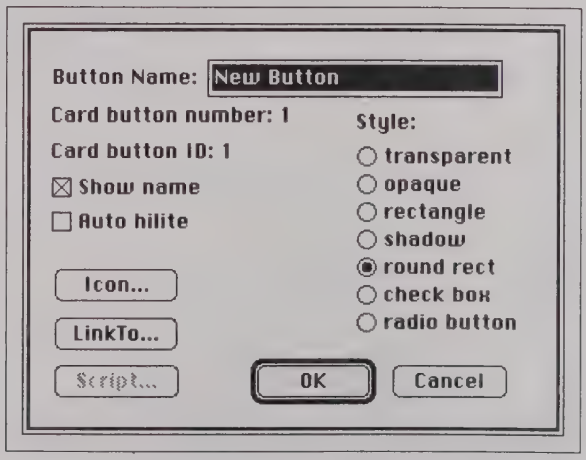
Nothing happens.

> HyperCard buttons work only when clicked by the Browse hand tool; the Button tool is used to create and change buttons, not to click them.

But in this case, switching to the Browse tool wouldn't help, because we haven't told the button what to do when it's clicked. We're ready to do that now.

Linking the Button

☐ **Double-click on the New Button button.**



This dialog box is what you'll use to "program" the new button. It allows you to name the button something besides New Button, change what the button looks like, and establish what the button will do when clicked.

Let's start by changing the name of the button.

- **Type "Click here to see inside".**

Since the original name of the button, New Button, was already highlighted in black in the dialog box, that name is replaced by your typing.

We have several choices of button styles; for this button, the already-selected round rectangle style is just fine. But it might be nice if we could have the button flash black when clicked, like official Macintosh dialog box buttons do.

- **Click on Auto hilite.**

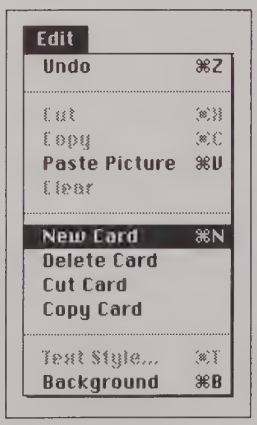
Now you're going to **link** this button to a new card showing a cross section of the pyramid, so that a click on that button transports the viewer to that cross-section card.

- **Click on LinkTo....**

A new window appears in the center of the screen:

This window is asking you to locate the card (or stack) that's to be linked to this button. In this case, that card doesn't yet exist, so we will have to create it.

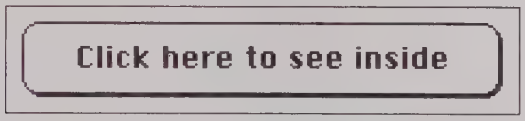
- **Select New Card from the Edit menu (or type Command-N, the keyboard equivalent).**



You're looking at a brand-new, all-white card, located right behind the first card in your stack. It's partially obscured by the small window, still waiting for you to tell it if we've reached the destination yet.

- **Click on the button labeled This Card.**

The window and the new card disappear, and you're back at the first card again, with a new name on the button.



Let's try the button to see if it works.

- **Press Command-Tab.**

This is the keyboard equivalent of selecting the Browse tool (the hand) from the Tools palette.

The new button is no longer highlighted.

- **Click on the button you created.**

If it worked properly, the button flashed black and the screen turned white; you've been transported to your new, all-white card. The Tools palette is still visible; it travels with you until you put it away. (If you're not there, you didn't link the button properly. You can remedy the situation by choosing the Button tool, double-clicking on the button, clicking on LinkTo..., pressing Command-4 to go to the last card in the stack—the new one—and clicking on This Card.)

Using the Clipboard

Now return to the first card so we can borrow from our previous artwork to decorate this card.

- **Press Command-1 (or select First from the Go menu).**

We'll make a copy of this picture with the Clipboard so we can use it as raw material for our next card. But before a graphic can be copied to the Clipboard, it must be selected. There are many ways to select a graphic in HyperCard.

HyperCard Graphic Selection Techniques

- To select the entire card's graphics, double-click on the **Selection rectangle** *or* use the **Select All command (Command-A)**.
- To select all of the graphic objects on the card without selecting the whole card, double-click on the **Lasso** *or* use the **Select command (Command-S)**.
- To select the most recently painted object (before you click anywhere else), do nothing (it's automatically selected) unless you want to move it; in that case, use the Select command (Command-S).
- To select a rectangular area, use the Selection rectangle.
- To select an irregularly shaped area, use the Selection rectangle with the Option key held down *or* use the Lasso *or* use the Select command after selecting an object with the rectangle.

Note: We'll modify this list in the next session when we work with backgrounds.

We'll use the Selection rectangle momentarily, and the Lasso in the next session. For our current job, the most appropriate tool is the Select All command.

☐ **Type Command-A (or choose Select All from the Paint menu).**

A dotted line should be moving around the edge of the screen, indicating that the whole picture is selected.

☐ **Type Command-C (or select Copy Picture from the Edit menu).**

You can't see it, but you just placed a copy of the entire drawing on the Macintosh Clipboard.

☐ **Press Command-~ (Go Back) to return to the blank card.**

When you're using the paint tools, the Go Back key means Undo the last step when typed by itself. To use it to go back one card, press the Command key with it. (This combination always takes you back, whether you're painting or not.)

☐ **Type Command-V (Paste Picture).**

The entire pyramid portrait should appear on the card, ready to be revised into a cross-section view. (Notice that the button did not copy with the artwork.)

The first thing we want to do is eliminate the title from this page so we have room for our subterranean passages.

- **Choose the Selection rectangle (the dashed rectangle) from the Tools palette.**

When you drag a diagonal with the Selection rectangle, you're selecting the rectangular area determined by that diagonal.

- **Drag a diagonal from the top right corner of the title to the bottom left corner, making sure to include the whole title in the rectangle. If you miss, click outside the selected area and try again.**

You should see a moving dotted line around the area you've selected. (Because of these moving dashes, the Selection rectangle has been nicknamed the marquee, or more informally, marching ants.)

- **Press the Delete (Backspace) key to delete the selected portion of the picture.**

That leaves a white space because you deleted some sand, too.

- **Select the Paint Bucket and click once in the white rectangle to fill it with sand paint.**

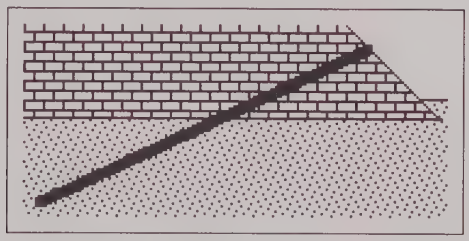
Painting Passages

- **Double-click on the Straight Line tool.**

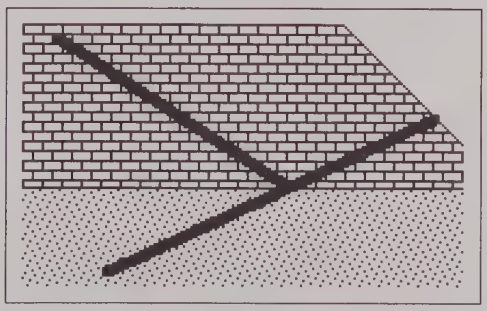
This is the shortcut equivalent of selecting **Line Size...** from the Options menu. You're going to be drawing some underground passages, and you'll need a line that's thicker than the one you've been using.

- **Click on the fourth line in the dialog box:**

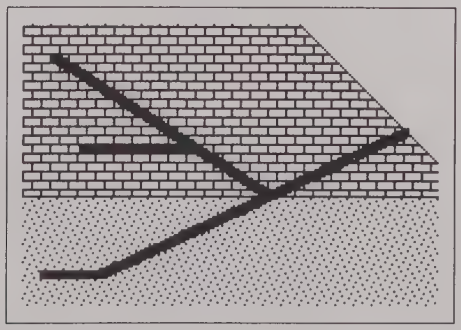
- Select the Straight Line tool and drag a line that looks like this:



- Now add another:



- Hold down the Shift key to constrain the Straight Line tool and draw two horizontal lines:

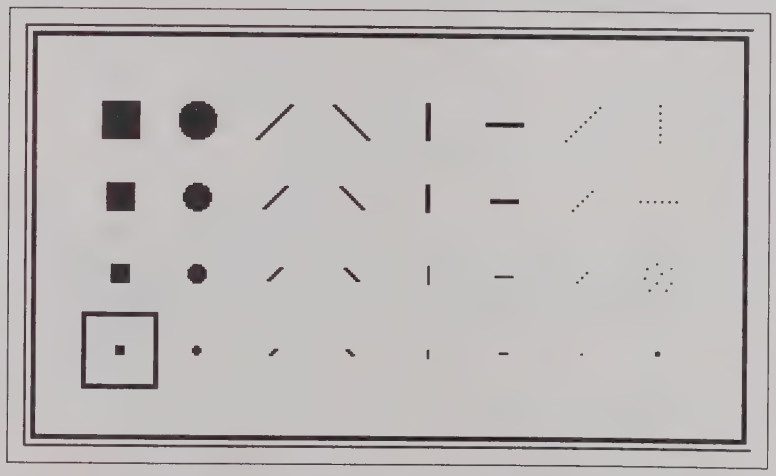


With the straight passages drawn, you're ready to draw the crooked escape passage.

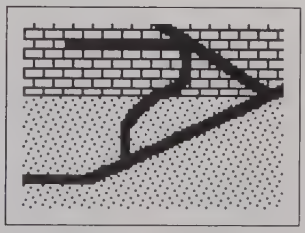
- ☐ **Select the Paintbrush tool.**
- ☐ **Select black paint from the Patterns palette.**

Double-click on the Paintbrush tool. Another keyboard shortcut, this one has the same effect as choosing **Brush Shape...** from the Options menu.

- ☐ **Select a brush shape from the dialog box (you *must* select one, even if it's the one that's already selected).**



- ☐ **Use the Paintbrush to draw the curved passage. If you make a mistake, type Command-Z (Undo) and try again.**



Now we can add the chambers.

Drawing Rooms

- **Select the Rectangle tool from the palette.**

When you drag between two points with this tool, you create a rectangle with those two points as opposite vertices.

- **Draw a rectangular-shaped chamber under the pyramid like this:**

Notice that the sides of the rectangle are the line thickness that we selected earlier; that thickness applies to all shape tools until we specify otherwise.

To be consistent with the black passages, we should paint the subterranean chambers black, too.

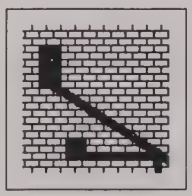
- **Select the Paint Bucket and the black paint.**
- **Place the paint drip *inside* the rectangle and click to pour.**

You can combine those two steps—creating a shape and filling it—by telling HyperCard that you want to draw filled shapes.

- **Double-click on the Rectangle tool.**

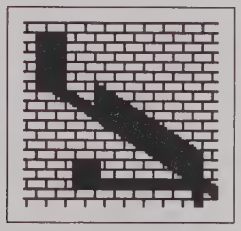
This is the same as selecting **Draw Filled** from the Options menu. Notice how the shape tools in the palette are now filled.

- **Select the Rectangle tool and draw two rectangles inside the pyramid, like this:**



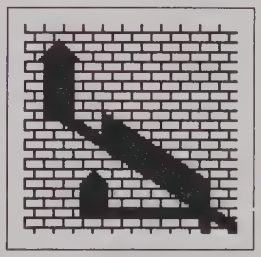
The Polygon tool also draws filled shapes.

- **Double-click on the Straight Line tool and select the thinnest line width.**
- **Select the Polygon tool.**
- **Draw a parallelogram like the one shown here by clicking around its vertices in order, starting and ending at the same corner.**



Notice how the shape fills with the selected paint when you complete it.

- **Add triangular peaks to the two internal chambers by clicking around the triangles with the Polygon tool.**



This is a good time to save your work, just in case.

- **Type Command-K (Keep).**

This picture clearly needs some labels.

Drawing Labels

- **Select the Rounded Rectangle tool.**

This tool works like the rectangle tool except that the corners are rounded. We'll use it to create labels. The labels should be filled with paint to obscure the pyramid bricks, but black paint would not be appropriate.

☐ **Select white paint from the Patterns palette.**

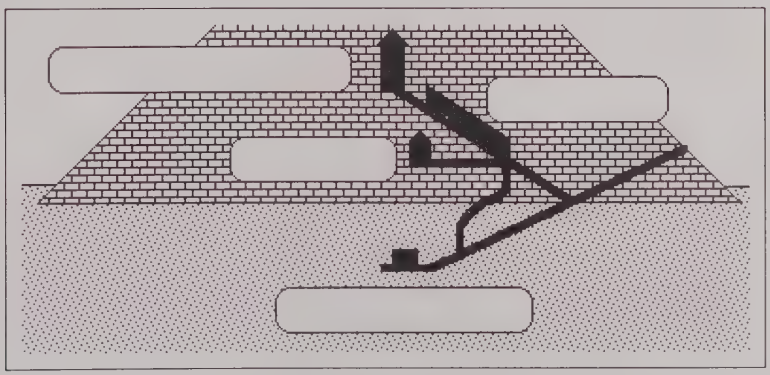
Even with the benefit of the HyperCard shape tools, it's difficult to make two shapes the same height. We can ensure that all of our labels are the same height, though, by forcing their edges to fall into some invisible grooves on the screen.

☐ **Select Grid from the Options menu.**

You'll notice the effect of the invisible grid as soon as you move the mouse around the card. Notice how the cross hairs jump from point to point rather than gliding smoothly across the screen.

> The grid is much more effective than the human eyeball for aligning objects on the screen. When you turn on the grid with the **Grid** command, it stays on until you turn it off by selecting Grid again.

☐ **Draw four rounded rectangles like these:**



In addition to noticing the effect of the grid, note how the white paint fill obscures the pyramid bricks and the sand.

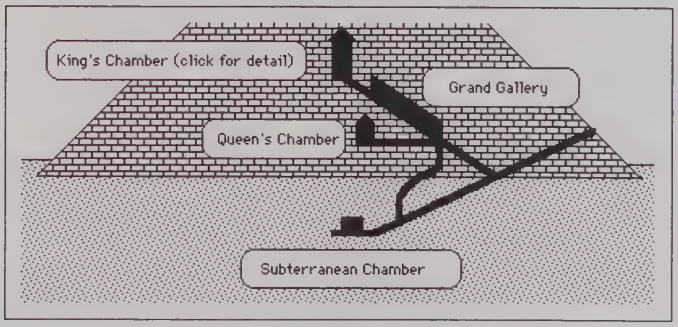
> There's a critical difference between white paint and no paint in a shape: A shape filled with white paint hides what it covers; an unfilled shape is transparent.

Now we can fill those shapes with text.

- **Double-click on the Paint Text tool (the letter A) to bring up the Text Styles dialog box.**
- **Select Centered, 9-point, Geneva with all other styles turned off. Click OK.**

This combination will apply to all of the text you type until you change it again.

- **Select the Paint Text tool, move the I-beam to the center of one of the rounded rectangles, click to position the pointer, and type the appropriate label. Repeat for the other labels in this picture:**



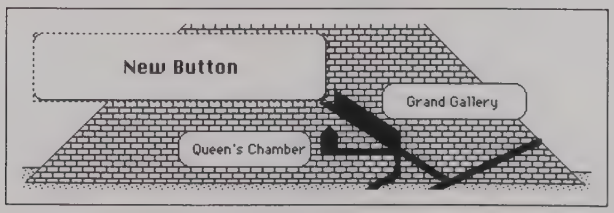
Creating an Invisible Button

Now let's provide our audience with a closer view of the King's Chamber.

- **Click on the Button tool in the Tools palette.**

New Button

- **Select New Button from the Objects menu.**
- **Drag this button so that it sits right on top of the King's Chamber.**
- **Stretch the corner so that it covers the chamber and the corresponding label completely.**



- [ ] **Double-click on the button to bring up the Button Info dialog box.**

We want the button and its name to be invisible.

- [ ] **Click on transparent button style.**
- [ ] **Click to turn *off* Show name.**

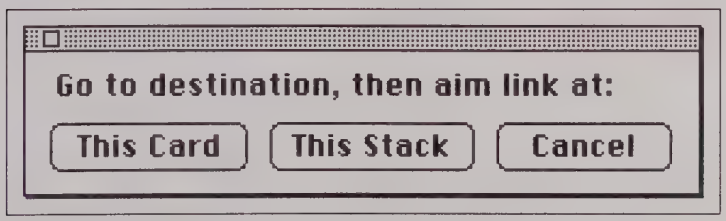
Since the name won't show on the screen, it's not strictly necessary to change its name. Do it anyway.

> It's a good idea to name buttons and other objects, even if those names won't be visible to the user. Those names just might make it easier for you to figure out what's going on if you decide to change something later.

- [ ] **Type "King's Chamber".**

Now it's time to link the button to a new card.

- [ ] **Click on LinkTo....**



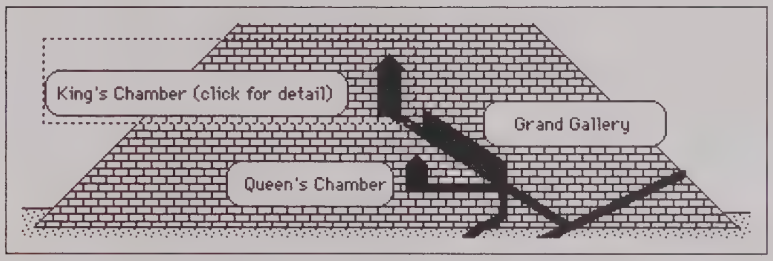
As before, we'll create a new card.

- [ ] **Select New Card from the Edit menu.**

Another white card will appear.

- [ ] **Click on the button labeled This Card.**

When you return to the cross-section card, your button is visible only in outline form.

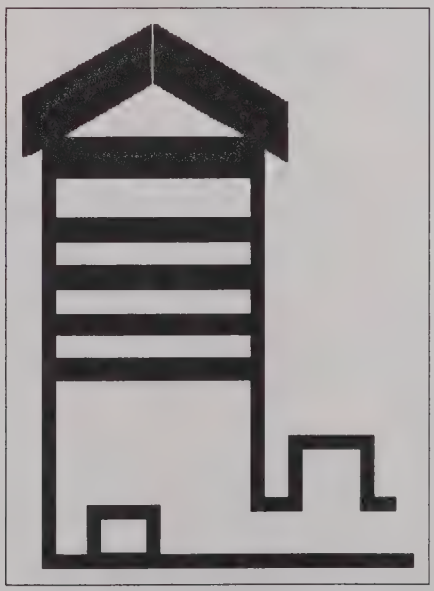


When you choose the Browse hand tool, the button will disappear altogether, but you'll know it's there because it will transport you to the new card when you click on it.

☐ **Type Command-Tab (or click on the Browse tool) and click on the King's Chamber.**

Drawing One Last Card

Now you need to fulfill the promise of the previous card by drawing a close-up view of the King's Chamber. Here's a sketch of what it might look like.



☐ **Use the Rectangle tool, the Polygon tool, the Straight Line tool, and the Eraser to draw a rough facsimile of this sketch on the blank card.**

Software Engineering Simplified

Software engineering is a branch of computer science that concerns itself with applying the design principles of engineering to the art and science of computer programming. When an engineer designs a bridge, we can generally assume that it won't fall down when we walk across it; when an engineer designs a TV, it's probably going to be easy enough to operate that we won't need to read a 300-page manual to use it. Similarly, if a computer program is well engineered, we should be able to assume that it works reliably and that it has a **user interface** that's easy to understand.

Unfortunately, most software isn't well designed when measured against these criteria. But there's no reason we can't apply some basic software engineering principles to the stacks we create to make them reliable and easy to understand.

The surest way to determine the reliability of the stack's performance is to test it thoroughly while you're building it and after it's completed. We've been testing as we go, but you should still go back and try all the buttons in different orders when the project is completed. This kind of exhaustive testing isn't always possible with large, complex stacks, so we'll look later at other ways to ensure stack reliability.

Designing an intuitive and functional user interface is a process that generally begins long before you sit down in front of the computer. But for this stack we'll simply look at the existing design and ask ourselves the question, "If I were a naive user with little or no HyperCard experience, what problems would I encounter with this stack?"

Adding Emergency Exits

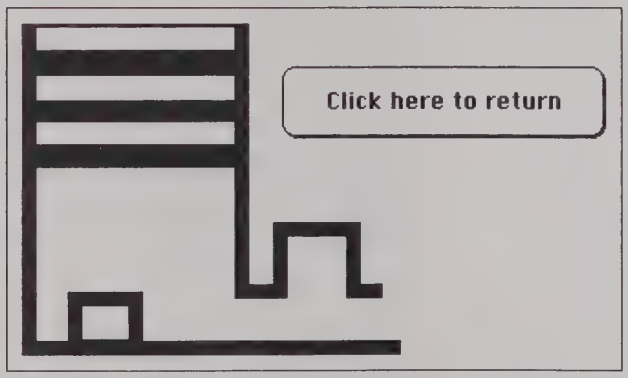
Of course, we'll want to look at the screens as we answer this question. But it's also helpful to examine a flowchart, or **map**, of the stack:

This diagram reveals a major design flaw: The stack is like a one-way street with a dead end; buttons lead you into the King's Chamber, but there's no button to take you out.

> Always give the user at least one exit from each card.

Let's add a button to return to the first card in the stack.

- **Select the Button tool.**
- **Select New Button from the Objects menu.**
- **Double-click on the button.**
- **Type "Click here to return".**
- **Click on the LinkTo... button.**
- **Type Command-1 to go to the first card in the stack.**
- **Click on the This Card button to complete the link.**
- **Change the shape and location of the button so it looks like this:**



Now let's test the button.

- **Select the Browse tool (Command-Tab).**
- **Click on the new button.**

You should be back outside the pyramid on card 1.

Borrowing a Button

We're almost through; all that remains is to provide the naive user with a graceful way to exit the stack and go Home. We need another button, but we won't create this one from scratch.

Address

- ☐ **Type Command-H (Go Home).**
- ☐ **Click on the Address icon to open the address book stack.**
- ☐ **Select the Button tool from the Tools palette.**
- ☐ **Click on the Home button to select it.**
- ☐ **Type Command-C (or select Copy Button from the Edit menu).**

Because the Button tool is selected, the Copy command now says Copy Button instead of Copy Picture. You now have a copy of the Home button on the Clipboard.

- ☐ **Press Command-~ twice to Go Back two cards, so you're looking at the pyramid again.**
- ☐ **Type Command-V (or select Paste Button).**

A shimmering copy of the Home button, complete with icon, should appear in the same position on the screen as the original.

- ☐ **Drag this button to the bottom right corner of the window.**

This is not just a lookalike button; it comes complete with instructions to make it do exactly what the original did: Take you home. Let's test it.

- ☐ **Select the Browse tool (Command-Tab).**
- ☐ **Click the new Home button.**

You're home! Now let's go back and install similar buttons on the other cards in the stack so our users have a convenient exit at any level.

- ☐ **Press Command-~ to return to your stack.**
- ☐ **Select the Button tool.**
- ☐ **Select the Home button you just added and press Command-C to copy it to the Clipboard.**

Actually, we already had a copy of the button on the Clipboard, but not in the right position.

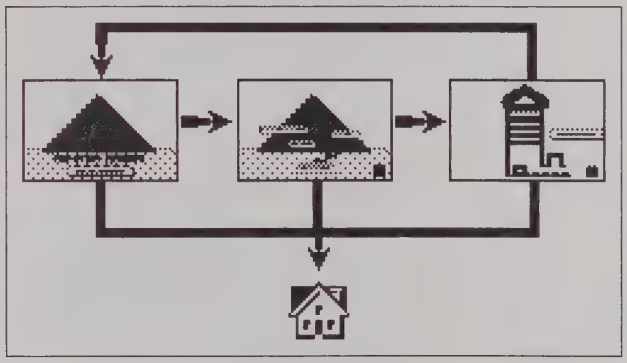
> When a button is copied from a card to the Clipboard and pasted on another card, it appears in the same relative location on both cards (until you move it).

- ☐ **Press Command-3 to go to the next card in the stack.**
- ☐ **Press Command-V to paste another copy of the Home button onto this card.**

It should appear in just the right spot. You still have one card in your stack without a Home button.

- ☐ **Type Command-3 to go to the next (and last) card in the stack.**
- ☐ **Paste another Home button here.**

Here's a map of the stack as it now stands:



There's plenty more we could do to this stack: We could use the Eraser, the Paint Bucket, and the Pencil to soften the lines at the base of the pyramid; we could use the Spray Can with sand-patterned paint to give the pyramid a weathered look; we could add buttons to the first card to reveal interesting facts, theories, and mysteries concerning the pyramid; we could add buttons to the cross section to zoom in on the other chambers of the pyramid—the possibilities are limited only by our creativity and time.

But that's enough for now. Take a couple of minutes to try out the stack, making sure that you understand what it's doing each time you click a button.

We've covered a lot of ground in this session. The tools and procedures you've used to build this stack can be used to build dynamic business charts, educational presentations, interactive tutorials, multilevel maps, and all kinds of other stacks. The exercises at the end of this session will suggest several projects that you might want to try on your own to practice your new skills. In the next session you'll learn more HyperCard tricks that will make it even easier to build your own stacks.

Summary

In this session we've focused on the fundamentals of stack construction. You learned how to reset the user level to authoring so that the stack construction tools are available, and you found out how to use many of those tools. You learned how to create new stacks and new cards. You used the Pencil, Straight Line tool, shape tools, Paint Bucket, Paintbrush, and Patterns palette to draw pictures on cards. You used the Clipboard, the Undo command, and the Eraser to edit those pictures. You practiced using paint text in a variety of fonts, styles, sizes, and alignments. You learned how to create buttons, label them, size them, position them, and link them to other cards. Finally, you learned how to copy buttons from other stacks into your own stack. Nice work!

Key Words

alignment (text)
authoring (user level)
Auto hilite (button)
bit
bit-mapped graphics
Brush Shape... command
constraint (Shift) key
Copy Button command (Command-C)
Copy Picture command (Command-C)
Cut Picture command (Command-X)
Cut Button command (Command-X)
Draw Filled command
Eraser tool
field text
font
general tools
Grid command
I-beam
Keep command
Lasso
Line Size... command
link
Link to... (button)
map
New Button command
New Card command (Command-N)
New Stack command
Objects menu
paintbrush tool
Paint Bucket tool
painting (user level)
paint text
Paint Text tool
paint tools
Paste Button command (Command-V)
Paste Picture command (Command-V)
Patterns menu
Patterns palette
Pencil tool
pixel
point (size)
Polygon tool
power keys
Rectangle tool
Revert command

Rounded Rectangle tool
Select All command (Command-A)
Select command (Command-S)
Selection rectangle
Show name (button)
size (text)
software engineering
Straight Line tool
style (text)
tear-off menu
Text Style... command (Command-T)
Tools menu
Tools palette
transparent button
user interface

Self-Testing Exercises

1. Under what conditions are the Paint, Objects, and Patterns menus visible?
2. What's the best way to draw a perfect square in HyperCard?
3. While putting the finishing touches on your drawing of Mom, you sneeze and make her ear much too big. Name two ways you can erase the offending line.
4. What's the fastest way to erase everything you've painted on the current card?
5. Describe how you can use the Clipboard to

   a. duplicate a button on a different card

   b. duplicate a portion of a picture on a different card

6. What does each of these standard HyperCard tool icons represent?

   a.

   b.

   c.

   d.

   e.

   f.

   g.

   h.

i.

j.

k.

7. Many of the tools in question 6 have a special meaning when double-clicked. Which special meanings can you remember?

Projects

1. Draw a tree chart showing the organization of the top levels of government. Your chart should contain a box labeled U.S. Constitution with three subordinate boxes representing the three government branches (executive, legislative, judicial). Create buttons for each of these subordinate boxes and link those buttons to cards that illustrate, describe, or subdivide those branches. Add a Return button to each new card and a Home button to the main chart card.

2. Draw a map of a baseball field with buttons on each fielder's position linked to "baseball cards" for each of these fielders on your favorite team. Include Return buttons on each card and a Home button on the baseball field card.

3. Draw a chart representing at least three generations of your family tree. Add a button to each node, and link each button to a card containing a description of that family member. Include a Return button on each card and a Home button on the main tree card.

4. Draw a hyper-greeting card. On the first card of the stack, draw the outside of the card, complete with graphics and the first part of the message. Add a button with a "click here" message and link it to the inside of the card containing the punch line and signature. Now go back and modify the original button's message and add a second button to the outside of the card offering the recipient an alternative message ("click here if you're male" on one; "click here if you're female" on the other, "click here if you're over/under 40"; "click here if you want a plausible excuse/the truth", and so on). The two buttons should link to different inside cards with different messages; both of these cards should have Home buttons on them.

5. Draw a map of your house, school, or dormitory. Place a transparent button on your room that will reveal a close-up view of your room. Add a Return button to the close-up and a Home button to the original map. Variations: Draw expandable maps of your city, your state, your country, the world, the solar system, or the galaxy.

SESSION 4

Objectives

By the end of this session you should be able to

- Use all of the HyperCard paint tools and understand how the effects of many of these tools can be modified using menu commands and various keystroke combinations
- Copy graphics from clip art and other sources for use in your stacks
- Use HyperCard's background layer to give cards in your stack a consistent look
- Create a stack with multiple backgrounds
- Design and build a complex network of cards to simulate a real-world environment

4

HYPERCARD'S HIDDEN LAYERS:

BACKGROUNDS AND BACKTRACKING

The Problem

Your cousin Andy—the smart one—is spending tomorrow afternoon at your house. You've got no time for playing games with him; you're busy learning HyperCard. But you expect he'll leave you little choice unless you provide him with something interesting to do. What you'd like to do is throw Andy in a dungeon that's full of so many rooms and passages that it would take him all afternoon to find his way back to you. Perhaps HyperCard can help.

Introduction

You know how to build stacks now, but you still haven't learned about some of HyperCard's most impressive and useful stack-building tools. The paint box is full of surprises, both fun and practical. And all of the paint tools become doubly useful once you learn how to take advantage of HyperCard's multiple layers.

Every card has two layers—a transparent **card layer** and a **background layer**—and you can put text, graphics, and buttons on either layer. This simple idea is one of HyperCard's most powerful concepts. Unfortunately, many beginners also find it to be one of HyperCard's most puzzling concepts. But as you work through this session's project, you'll find that those layers are crystal clear—and convenient, too.

This session's project is more complex than the last one. Consequently, there are more ways that things can go wrong. So we'll spend some time looking at ways we can design the stack that will minimize the chances of errors.

Some Background on Backgrounds

Let's start by looking behind the scenes at Home to see how backgrounds work in a familiar environment.

Examining Home's Background

- ☐ **Turn on your computer. If you're using a hard disk, boot from the hard disk before inserting your Home disk. Otherwise, insert your system/Home disk(s) to boot the computer.**
- ☐ **Double-click on the Home stack icon.**
- ☐ **Choose Background from the Edit menu.**

(If your Edit menu doesn't have a Background command, your user level is probably not set at authoring; change that on the last card of the Home stack as before.)

You should see two changes:

1. All of the text and icons have vanished from the Home card, except for two lonely buttons at the bottom.
2. The menu bar has grown candy **stripes**.

The stripes are there to remind you that you're looking, not at a card, but at a background. In this case, the background is the one used by

every card in the Home stack. It includes the index-card border and the left- and right-arrow buttons, and nothing more.

- ☐ **Select Background from the Edit menu again, or use the keyboard shortcut, Command-B.**

Browse through the Home stack once again, noticing how every card in the stack shares those common background elements.

> **The Benefits of Backgrounds**
>
> In most stacks, the same background is shared by several, sometimes all, of the cards. There are practical and aesthetic advantages to designing a stack this way:
>
> - It saves stack creation time, because you don't need to create or copy common elements on many different cards; you can build them once in the common background.
> - It saves space on disk and in memory, because the stack isn't filled with redundant elements.
> - It usually results in a more consistent user interface, which helps users to find their way around. (For example, in the Home stack, the two arrow buttons are in exactly the same place on every card.)
> - It gives the stack a more consistent, appealing look.
>
> Many stacks, such as the DateBook stack, have different backgrounds for different groups of cards. The advantages of backgrounds apply to those stacks, too.

When we built the Pyramid stack, we left the background plain white and focused on the card layer. In this session we'll create a stack with several different backgrounds to maximize our efficiency.

A HyperCard Adventure

Let's build a HyperCard dungeon to occupy cousin Andy. The dungeon will have all kinds of rooms linked together by doors, windows, and secret passages . . . and only one way out.

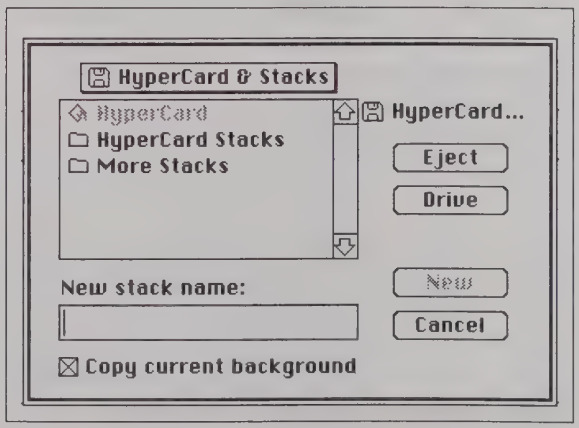
This is one variation on a popular kind of computer game known as an **adventure game**. Some adventure games are text-based **interactive fiction** in which the reader types commands to tell the main character what to do. Others, like the one we'll create, use graphics instead of text to show the setting and develop the plot of the game. Whatever its form, an adventure game can be thought of as a computer **simulation** of a real or not-so-real environment and situation.

A graphic adventure game is a perfect project for HyperCard; each room can be represented on one or more cards in a stack; hidden buttons can transport the player to other rooms. To save time and minimize duplication, we can create generic rooms in the background layer and fill in the details on the card layer.

Starting a Stack

Before we get started, it's only fair to warn you: Creating a maze can be as confusing as navigating a maze. If you don't pay attention and carefully follow every step of this process, you may become as lost as your intended audience. Mark each step in this session as you complete it so you can be sure that you don't skip any steps. Focus on understanding the structure of the stack more than on making the graphics perfect. If you're working on a complex graphic, use Keep (Command-K) regularly to save your work in progress, so you don't risk losing your train of thought while you reconstruct a spoiled picture. Pay attention to whether you're in the card layer or the background layer throughout the session. If you don't understand why you're doing something, read ahead to find out. If you lose your bearings, backtrack through the steps until you find the point where you wandered off the path. If, in spite of everything, your stack becomes hopelessly botched, don't despair. If you don't have time to start fresh, simply read through the rest of the session, focusing on understanding the underlying concepts.

☐ **Select New Stack from the File menu.**



Make sure the disk shown in the window is your disk.

- ☐ **Type "Dungeon".**
- ☐ **Click on the check box to turn *off* the option labeled Copy current background.**

We'll be creating our own backgrounds, thank you.

- ☐ **Click New.**

We now have the simplest stack of all: one card with a transparent card layer and an all-white background. Let's get our tools together and go to work on the background.

- ☐ **Tear off the Tools menu and place it in the upper right corner of the screen.**
- ☐ **Click on any paint tool to make the painting menus (Paint, Options, Patterns) appear.**
- ☐ **Drag the Patterns menu to the left side of the screen directly opposite the Tools palette.**

Drawing in the Background

- ☐ **Type Command-B (or select Background from the Edit menu).**

Every good adventure game needs an enticing introduction, preferably scrawled on a tattered scroll. So let's create a scroll on the background.

> It's easy to forget to change to the background layer, so always check the menu for stripes to be sure you're working in the background.

All of the paint tools work in the background just like they do on the card layer. Let's start by drawing one edge of our scrolled message:

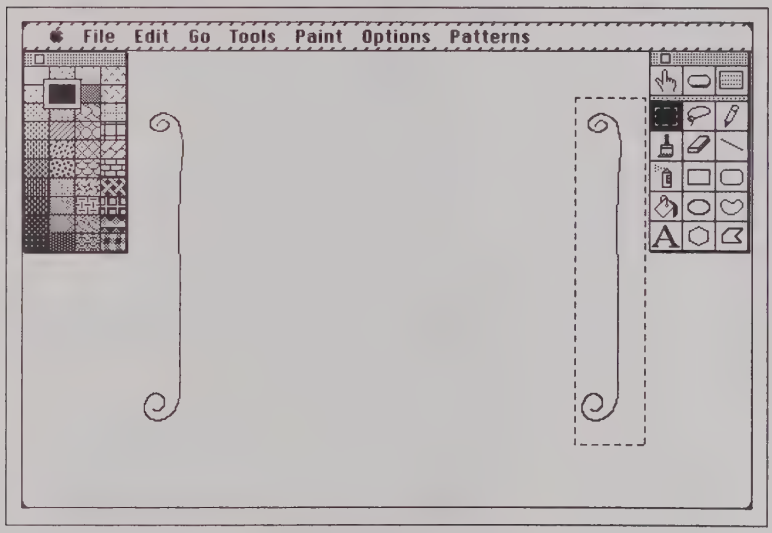
- ☐ **Select the Pencil and draw a 3-inch squiggly line something like the one at left.**

If you don't get it exactly right the first time, remember Command-Z, the Undo command.

- **Use the Selection rectangle to select the line and drag it until it's close to the Patterns palette but not hidden by it.**

That's the left side of the scroll; we need another line just like this one for the other side. We *could* Copy and Paste one, but we'll use the Option key to clone it instead.

- **Hold down the Shift and Option keys and drag the selected line until it's near, but not touching, the Tools palette.**



When Dragging a Selected Graphic Object

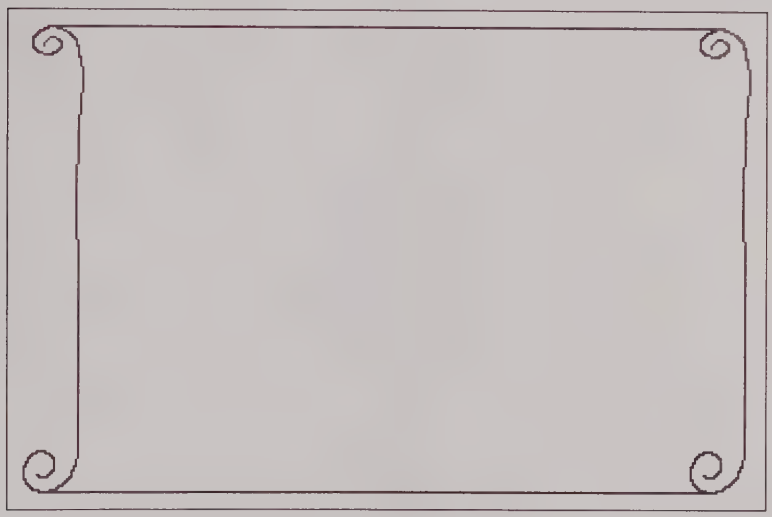
- The Shift key constrains the movement to horizontal or vertical only.
- The Option key leaves a copy behind.
- If you hold down the Command key and drag the edge of the object, the object doesn't move; it stretches or shrinks.
- If you hold down the Command and Shift keys and drag the *corner* of the object, the object stretches or shrinks while maintaining its original proportions.

Note: **Stretching** or **shrinking** bit-mapped graphics usually results in some distortion, especially if you don't maintain proportions.

We have the sides; now let's draw the top and bottom of the scroll.

☐ **Use the Straight Line tool to connect the tops of the two lines.**

Remember the trick from the last session: If you hold down the Shift key when you drag between two points, you'll find that it's a cinch to draw perfectly horizontal (or vertical) lines.



It's a start, but it's still not a scroll.

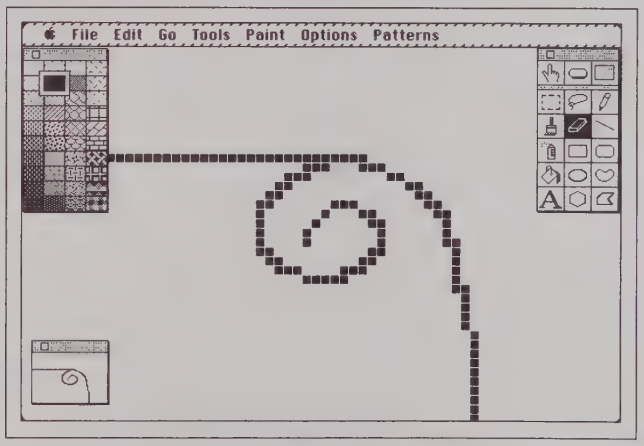
☐ **Add straight lines to the two left scroll corners, like this:**

Now we need to eliminate the curls from the right side of the scroll using the Eraser and the Pencil. The Eraser's large, square shape makes it awkward for that kind of fine detail work. We can't make the Eraser smaller, but we can make the details of the picture bigger.

- ☐ **Click once on the upper right corner of your scroll.**

You're telling HyperCard what part of the picture you want to focus on. Don't worry if you leave a mark there; you can erase it soon enough.

- ☐ **Double-click on the Pencil tool to zoom in for an extreme close-up.**



The Options menu command for doing the same thing is called **FatBits**. It's easy to see why, if you remember that each pixel (dot) on HyperCard's bit-mapped screen is represented by a bit in the computer's memory. In FatBits each of these bits is exactly eight times fatter than normal. The small window in the corner of the screen displays a normal magnification view of the same portion of your picture.

- ☐ **Use the Pencil and/or the Eraser to remove the unwanted curl from the picture.**

- ☐ **Hold down the Option key and use the hand to drag the magnified picture up so that you can follow the scroll edge down to the curl in the lower right corner.**
- ☐ **Remove the unwanted curl in the lower right corner of the scroll.**
- ☐ **If necessary, use FatBits to touch up the other two corners of your scroll, making sure that all of the lines connect.**
- ☐ **Exit FatBits by clicking in the small normal view window (or select FatBits from the Options menu).**

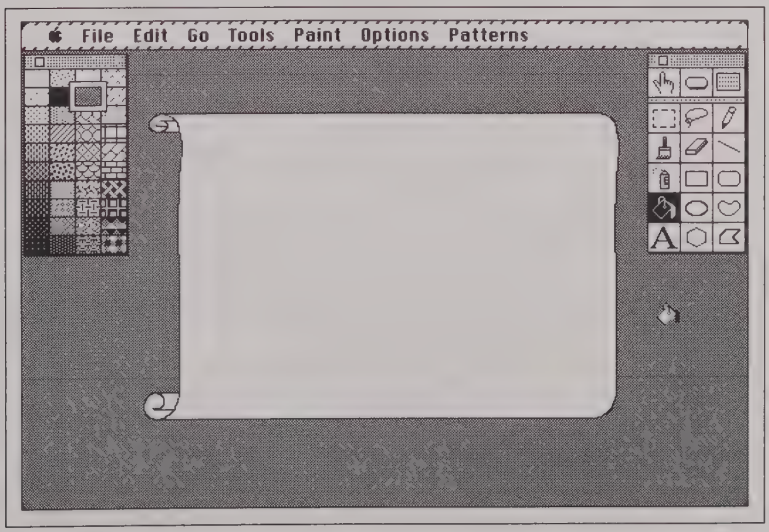
FatBits Tips

- The Pencil is probably the most useful FatBits tool, because it allows you to turn each pixel on or off with just a touch. As usual, the Pencil draws black if you start on a white pixel and white if you start on black.
- The Eraser is also helpful because it is much smaller relative to a pixel than it is in normal magnification.
- The other paint tools work in FatBits, but some may seem unwieldy at that magnification layer.
- If you hold down the Option key while in FatBits, the pointer turns into a hand that you can use to drag the magnified picture so the FatBits window reveals a different part of the picture.
- The picture is shown actual size in the small window.
- You can return to normal magnification by clicking on the small normal view window or by selecting FatBits again.

Let's add a background pattern to this background layer.

- ☐ **Select a medium gray pattern.**
- ☐ **Use the Paint Bucket tool to paint the area surrounding the scroll.**

Before you do anything else, check to make sure that only the area around the scroll was painted. If part of the scroll changed, you'll need to select Undo, fix the leak, and repaint the areas.



If you like, you can return to this background layer later to add some weathering and shading around the edges with the Spray Can and other tools, but for now, let's leave the background layer and add words and a button to the card layer.

- ☐ **Type Command-B (or select Background from Edit).**

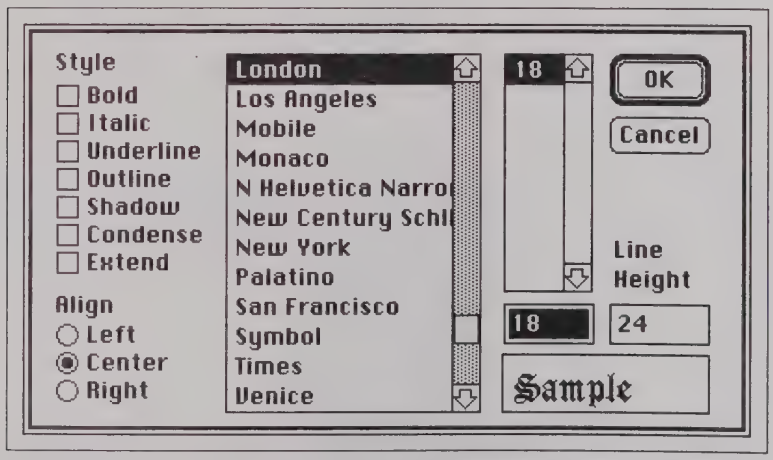
The only change you should notice is that the candy stripes vanished from the menu bar. We're now working on the card layer, but everything looks the same because *the card layer is transparent until you put something on it.*

Creating Titles

A

- ☐ **Double-click on the Paint Text tool to bring up the Text Styles dialog box.**

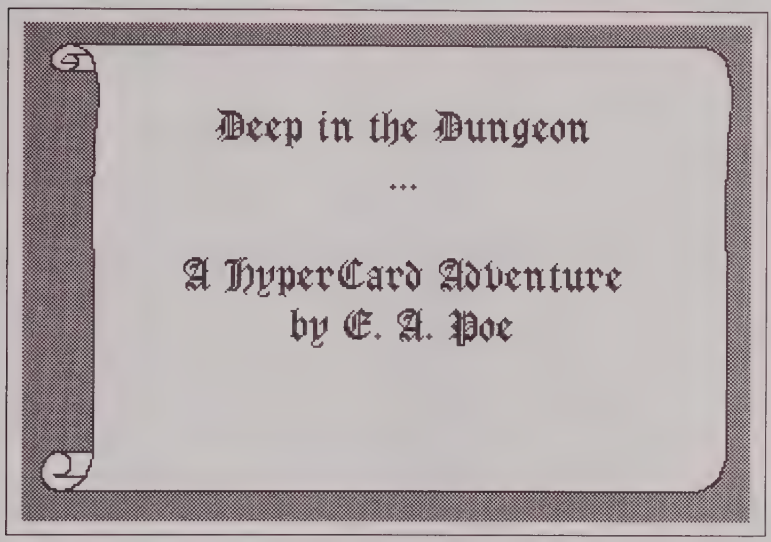
Ideally, we'd like to use a font that looks like it was drawn by a medieval scribe. Such fonts exist, but not in the standard Apple system package. We'll use London here, but you can choose any font that you have in your System in a large size. (Remember that the sizes stored in your System for a font are listed in the dialog box when you select that font.)



- ☐ **Select a font; then select the largest available size for your chosen font.**
- ☐ **Select Align Center and click OK.**

When you put away this dialog box, you'll find that the Paint Text tool is automatically selected; HyperCard is anticipating your needs.

- ☐ **Click about a half-inch from the top of your scroll near the center.**
- ☐ **Type something like this, pressing Return at the end of each line:**



Since you're working with paint text, you don't have access to all of the conveniences of field text. Specifically, you can't assume that the text will automatically wrap around to the next line when one line is full, and you don't have the wealth of text-editing tools that we used in the last session. If you make an error, your only recourse is to backspace to the offending characters and retype the remainder of the text. Once you reposition the insertion point or select another tool, your text is dried paint.

- ☐ **When you have typed your text, type Command-B to take another look at the background.**

The background doesn't contain any of the words you typed; it's still clean and available for use on your next card.

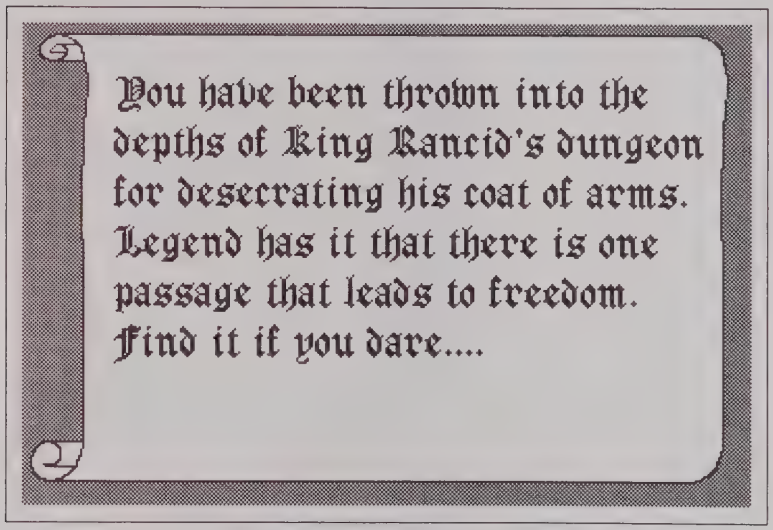
- ☐ **Type Command-N to create a new card.**

You're whisked out of the background layer and onto a new card that looks just like the old one, except that this one contains no text.

> When you create a new card with the New Card command (Command-N), the new card shares a background with the current card and is positioned directly after it in the stack.

Let's add some text to this card.

- ☐ **Double-click on the Paint Text tool, select Align Left, and click OK.**
- ☐ **Position the insertion bar near the upper left corner of the scroll and type the following, pressing Return after each line:**


You have been thrown into the
depths of King Rancid's dungeon
for desecrating his coat of arms.
Legend has it that there is one
passage that leads to freedom.
Find it if you dare....

Borrowing a Background Button

We have two introductory cards, but no buttons to link the cards to each other and to the game. We *could* put a button on each card specifically linked to the following card, just like we did in the last session. But since these two cards share a background, we could install one "Go to the next card" button in the background layer that will service both cards. We can borrow just such a button from the Home stack.

- ☐ **Type Command-H to go Home.**
- ☐ **Select the Button tool.**

- ☐ **Click on the right-arrow button.**

Recall that this button takes you to the next card in the stack no matter where you are. This is exactly what we want our button to do in our new stack.

- ☐ **Type Command-C to copy the button to the Clipboard.**

- ☐ **Type Command-~ to go back to your Dungeon stack.**
- ☐ **Type Command-B to go to the background.**
- ☐ **Type Command-V to paste the button.**
- ☐ **Drag the button to a position near the bottom of the scroll.**

All we need to do now is change the look of the button so that it fits in with the medieval decor.

- ☐ **Double-click on the button to open the Button Info dialog box.**
- ☐ **Double-click on the button labeled Icon... in the dialog box.**

- ☐ **Scroll through the icons, select an appropriate icon for the period, and click OK.**
- ☐ **Type Command-B to leave the background.**

- ☐ **Type Command-Tab to select the Browse tool (the pointing hand).**
- ☐ **Click on the button a few times to try it.**

We've changed the look, but not the function, of the button. It takes us to the next card in the stack, whatever that might be.

> **Absolute and Logical Links**
>
> A button created with the Link To... procedure is hard-wired to a particular card by an **absolute reference**. A button that says "Go to next card" uses a **logical reference** to link to the next card, whatever it might be. Computer spreadsheet users deal with a similar distinction every day when working with formulas in cells. So do you, unconsciously, when a stranger asks you, "Where is the office for the local newspaper?" The absolute answer, "435 Madison Street," is helpful if your questioner wants to send letters that will reach the editor from any mailbox in town. The logical answer, "six blocks south" or "the end of the street," is more helpful if the questioner has an appointment to meet with the editor today.
>
> Each of these reference types has advantages and disadvantages. Absolute links (Go to this card; Go Home) take you to the specified destination card no matter how many or which cards are in between. Logical links (Go Next; Go Last) allow you to add cards to or remove cards from a sequence without having to relink the cards on either side. A logical "Go Next" button installed in the background can replace many absolute "Go to this card" buttons installed on the card layer. You'll see how each of these link types can be used in other situations as you progress through this book.

Creating a New Background

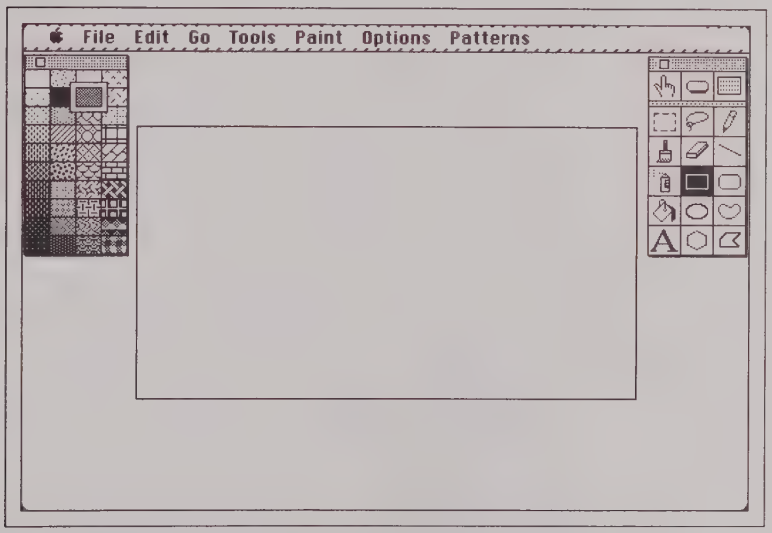
Our next card will be the first room of the dungeon. If we simply select New Card, the card we create will have the scroll background. We need to tell HyperCard to create a new card with a new background.

- ☐ **Go to the last card in the stack by typing Command-4.**
- ☐ **Select New Background from the Objects menu.**

The menu bar is striped to indicate that we're in the background—a new background that we'll use for the next few cards. We want to create a sort of generic dungeon room that can be customized on the card layer for different rooms in the dungeon.

The **New Background** command creates a new card with a new background directly after the current card.

- **Select the Rectangle tool.**
- **Double-check the menu bar to make sure you're in the background. Then draw a rectangle like this in the center of the screen:**

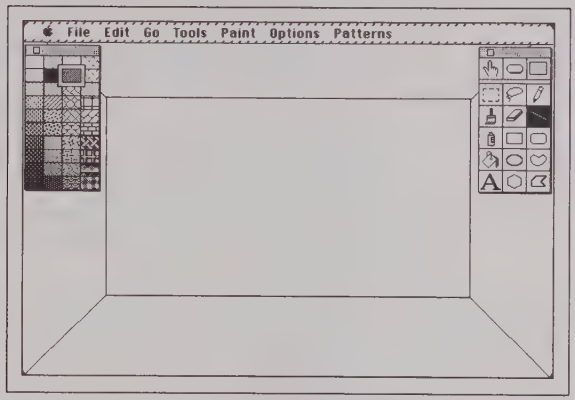


Use the Patterns palette and the Tools palette as guides when you're creating your rectangle. The exact size isn't important, as long as you do your best to center it.

- **Use the Straight Line tool to draw lines from each corner of the rectangle to the corresponding corner of the screen.**

For the top two corners, you'll need to drag the line behind the menu bar to the absolute corner of the screen.

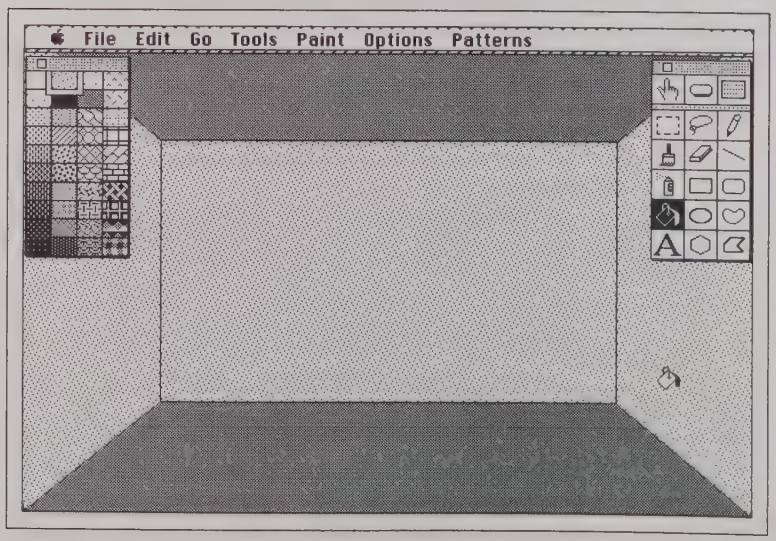
The top of a card may be obscured by the menu bar, but it's still there. You can type Command-Space to temporarily hide the menu bar so you can see what you're painting.



We now have three walls, a ceiling, and a floor, but they need paint.

- ☐ **Select the Paint Bucket tool and the medium gray paint pattern.**
- ☐ **Pour that paint into the bottom (floor) and top (ceiling) sections of your picture.**
- ☐ **Select an irregular dot pattern or a light gray pattern and paint the three remaining sections with the bucket.**

It's tempting to use the brick paint for the walls, but if we use that or any other regular pattern on the side walls, the illusion of perspective we've created will be shattered.



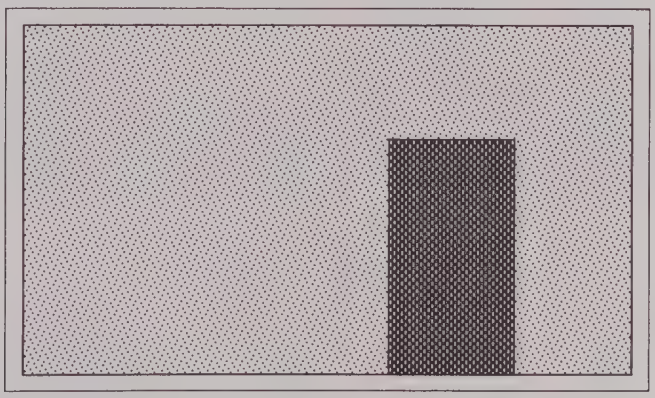
The generic room background is finished and ready to be customized on the card layer.

- ☐ **Type Command-B to leave the background.**

Shaping Doors and Windows

Let's start with a door.

- ☐ **Double-click on the Rectangle tool (or choose Draw Filled from the Options menu).**
- ☐ **Select a likely-looking pattern for the door.**
- ☐ **Select the Rectangle tool.**
- ☐ **Draw a door in the far wall of the room, like this:**

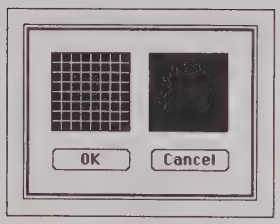


Notice how the door obscures the background.

> Objects drawn on the card layer hide anything behind them in the background layer (unless you specify that the object is transparent).

If this is truly a dungeon, it should have a barred window on the door. We could draw a window and then draw bars, but we can do a better job faster by creating a bar pattern.

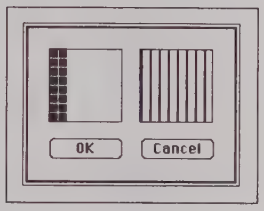
- ☐ **Double-click on the black pattern in the Patterns palette (or choose Edit Pattern... from the Options menu).**



The box on the left shows you the currently selected pattern close-up; the one on the right shows it at normal size.

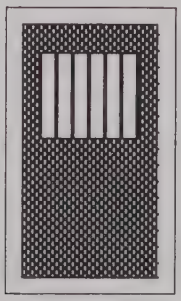
You can edit this pattern by clicking or dragging on the pixels in the left box; the results are immediately reflected on the right.

☐ **Draw the pattern shown here and then click OK:**



Notice how the original pattern has been replaced in the palette.

☐ **Select the new pattern and draw a rectangle in the center of the door like this:**

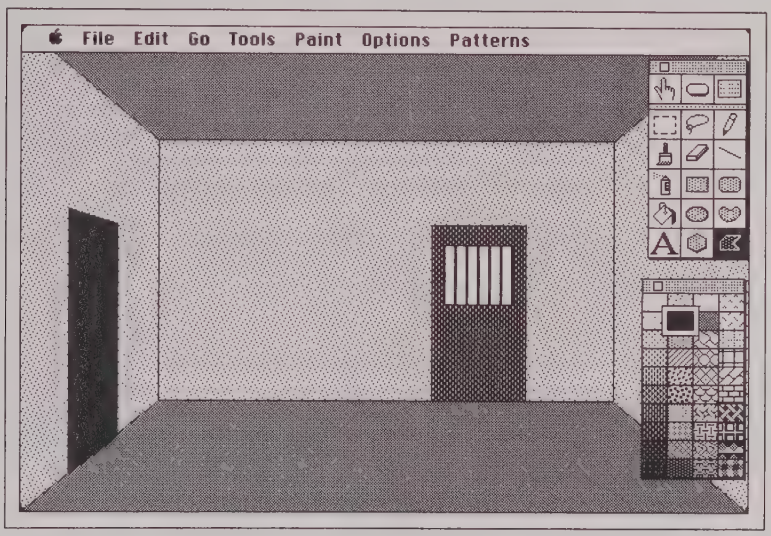


Now, in the spirit of covering our tracks, we should put that pattern back the way we found it.

☐ **Double-click on the bar pattern and edit it so it's all black again. Click OK.**

Let's add one more door to the side wall.

- ☐ **Move the Patterns palette below the Tools palette.**
- ☐ **Select the Polygon tool.**
- ☐ **Click on four points to define the polygon shown here, finishing the shape by clicking on the first point again.**

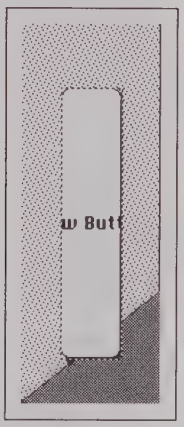


This will be the door into the other chamber of our luxury dwelling. We need a button to take us there.

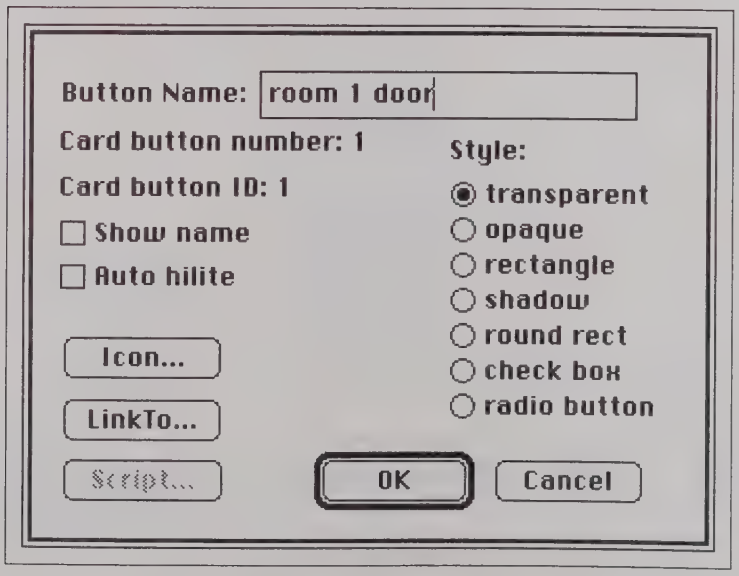
Creating a Two-Way Link

- ☐ **Select the Button tool.**
- ☐ **Select New Button from the Objects menu.**

- Drag and reshape the button so that it covers the door on the left wall:

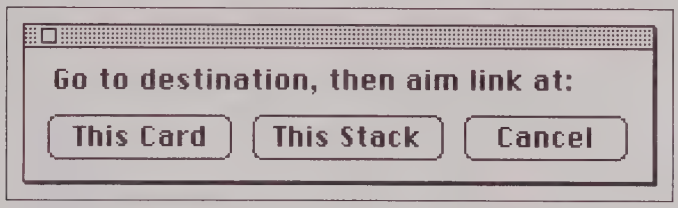


- Double-click on the button.



Follow these familiar steps to fill in the dialog box:

- ☐ **Click on the Show name checkbox to turn it off.**
- ☐ **Click on transparent.**
- ☐ **Type "room 1 door" (this step is optional).**
- ☐ **Click on LinkTo....**



- ☐ **Type Command-N (or select New Card from Edit).**

The new card has the generic room background and is located after the starting room card in the stack.

- ☐ **Click on the This Card button.**

You're back to the starting room. You might notice that the barred door looks different. Sometimes when you tell HyperCard to make the button transparent, it makes the graphics on the card layer **transparent**, too. This is not as it should be, but it's easy to fix.

- ☐ **Hold down the Command key while dragging the Selection rectangle around the door with bars.**

The Command key forces the Selection Rectangle to select the door and nothing more.

- ☐ **Select Opaque from the Paint menu.**

You can no longer see the background pattern through the door.

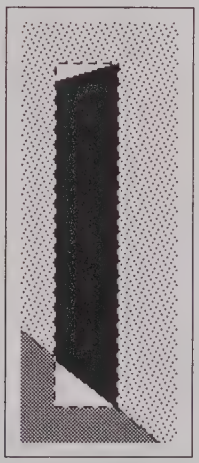
As long as we're back in this room, let's borrow the black door so we can make it visible in the new room.

- ☐ **Hold down the Command key while dragging the selection rectangle around the black door.**
- ☐ **Type Command-C (Copy Picture).**

- ☐ **Type Command-3 (Go Next).**
- ☐ **Type Command-V (Paste Picture).**

We've put the door into the next room, but it's backward.

- ☐ **Choose Flip Horizontal from the Paint menu.**
- ☐ **Move the palettes out of the way and drag the door until it's positioned in the center of the right wall.**



The white areas are there because the selected picture is **opaque**; you can't see the background through it.

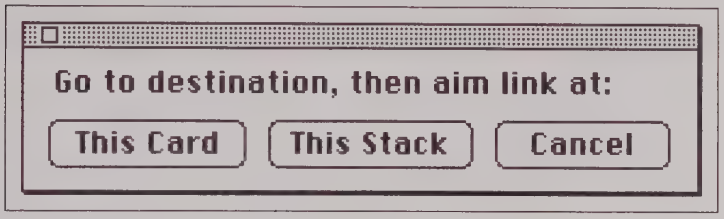
- ☐ **Select Transparent from the Paint menu.**

Now the door is visible from both sides of the wall. But as it stands now, it's a one-way trip. We need to place an invisible button on the backside of the door that links back to the first room.

- ☐ **Click on the Button tool.**
- ☐ **Select New Button from the Objects menu.**
- ☐ **Drag and reshape the new button so that it covers the door.**
- ☐ **Double-click on the button to bring up the Button Info dialog box.**
- ☐ **Click on Show name in the dialog box to turn it off.**
- ☐ **Click on transparent.**

- ☐ **Type "room 2 door" (this step is optional).**
- ☐ **Click on Link To....**

You'll see this familiar window, but you won't create a new card. This button should take you back.



- ☐ **Type Command-2 (Go Back).**
- ☐ **Click on the button labeled This Card.**
- ☐ **Type Command-Tab (Browse tool) and test the door both ways. Make sure you finish up in the latest room (with only one door).**

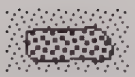
Building Portable Furniture

No good dungeon is complete without a secret passage. Let's build a bed and hide our passage underneath. We'll start by drawing a pillow.

- ☐ **Select a pattern for the pillow.**

- ☐ **Select the Curve tool.**

The Curve tool works pretty much like the Pencil, except that the drawn shape fills automatically if Draw Filled is selected.



- ☐ **Draw a pillow anywhere, so long as it doesn't touch the door.**

(We're on the card layer, and the door is the only other painted object on this layer. As long as we don't put the pillow on the door, we can move it where we want later.)

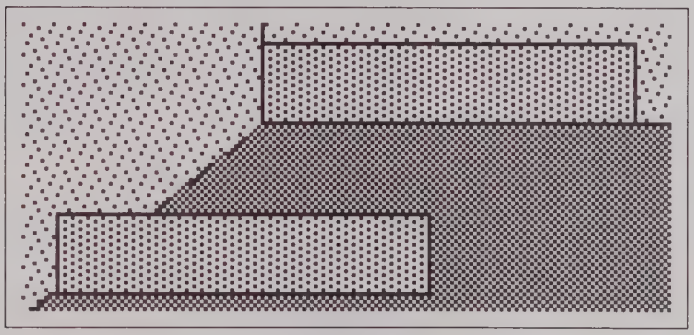
If you can see the background pattern through your pillow, it's because HyperCard thinks you want everything on this card layer to be transparent again. Here's how to fix that.

- ☐ **Double-click on the Lasso to select all graphics on this layer.**
- ☐ **Select Opaque from the Paint menu.**

(If you had double-clicked on the Selection rectangle instead of the Lasso, the entire area inside the rectangle, including the area between graphic objects, would become opaque, hiding the background completely.)

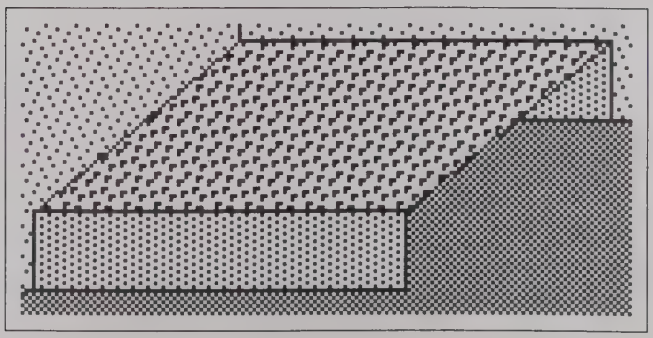
Now let's draw the bed.

- ☐ **Choose a pattern for the ends of the bed.**
- ☐ **Use the (filled) Rectangle tool to create a rectangle in the corner of the room, like the top one in this picture.**

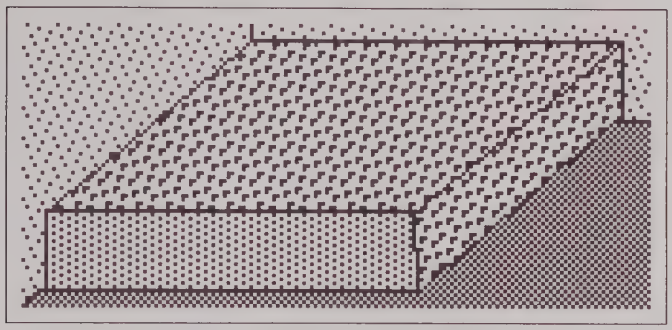


- ☐ **Choose Select from the Paint menu to select the new rectangle.**
- ☐ **Option-drag the rectangle to clone the other end of the bed.**
- ☐ **Select the Polygon tool.**
- ☐ **Choose a pattern for the bedspread.**
- ☐ **Click around the corners of the top of the bed ends to create the shape for the top of the bed.**

When you click on the first point a second time, the shape should fill.



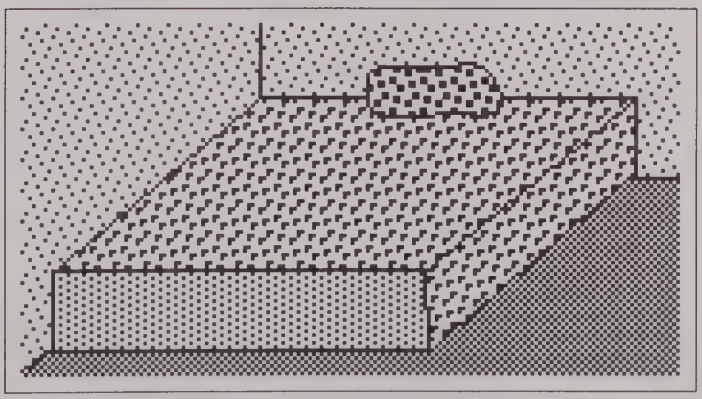
- ☐ **Click on the corners of the side of the bed to define its shape.**



- ☐ **Select the Lasso tool.**
- ☐ **Drag a line around the outside of the pillow without actually touching the edge of the it.**

You don't need to drag all the way around the pillow; if you go most of the way and let go, the two ends of your curve will be connected automatically. When you let go, every painted object that's within the Lasso line on the card layer is selected, but none of the empty space between or around the pillow is selected. The Lasso is a shrink-to-fit tool.

- ☐ **Drag the selected pillow onto the bed.**



The Lasso is a particularly useful tool for selecting irregularly shaped objects, but it's not the only way. Let's revise the list of selection techniques we started in the last session:

HyperCard Graphic Selection Techniques

- To select the entire graphics layer of the card or background, double-click on the Selection rectangle *or* use the Select All command (Command-A).
- To select all of the graphic objects on the card or background layer without selecting the whole layer, double-click on the Lasso *or* use the Select command (Command-S).
- To select the most recently painted object (before you click elsewhere), do nothing (it's automatically selected) unless you want to move it; in that case, use the Select command (Command-S).
- To select a rectangular area of the current layer, use the Selection rectangle.
- To select an irregularly shaped area on the current layer, use the Selection rectangle with the Option key held down, *or* use the Lasso, *or* use the Select command after selecting an object with the Selection rectangle.

Moving Furniture

If the secret passage is under the bed, we want to be able to move the bed with a click. We'll do that by linking a bed-sized button to a new card just like the current card, but with the bed in a different location. Repeat the now familiar steps for creating a linked button.

- ☐ **Select the Button tool.**
- ☐ **Select New Button from Objects.**
- ☐ **Drag and stretch the new button so it covers the bed, more or less.**
- ☐ **Double-click on the button to open the Button Info dialog box.**
- ☐ **Click on transparent.**
- ☐ **Click on Show Name to turn off that option.**
- ☐ **Type "bed" (optional).**
- ☐ **Click on LinkTo....**
- ☐ **Type Command-N (New Card).**
- ☐ **Click on the This Card button.**

When you return, the bed may be transparent.

- ☐ **Double-click on the Lasso to select all of the objects in this layer.**

- ☐ **Select Opaque from Paint.**

Since all of the graphics in the card layer are selected, you can easily copy them to the new card.

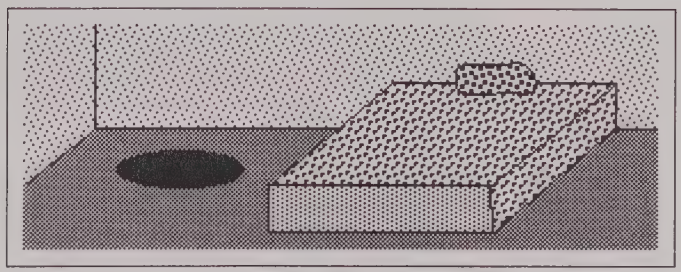
- ☐ **Type Command-C (Copy Picture).**
- ☐ **Type Command-3 (Next Card).**
- ☐ **Type Command-V (Paste Picture).**

This card now looks just like the other one. It should, except that the bed must be moved to reveal the secret passage.

- ☐ **Lasso the bed and drag it to the right a couple of inches.**

Where's the secret passage?

- ☐ **Select black paint and the Oval tool.**
- ☐ **Draw a black oval where the bed was originally.**



There's the secret opening, waiting for a button to link it to a new room or passage. Until that button is installed, the door is the only exit.

- ☐ **Type Command-Tab (Browse) and click on the door.**

That won't even take us out because we haven't installed a button *on this card*. We can fix that.

- ☐ **Type Command-2 (Go Back).**
- ☐ **Select the Button tool.**
- ☐ **Click on the Door button.**
- ☐ **Type Command-C (Copy Button).**
- ☐ **Type Command-3 (Go Next).**
- ☐ **Type Command-V (Paste Button).**

This button is an identical copy of the button from the previous card, so it should do just what the other was trained to do: Take us back to room 1. Try it.

- ☐ **Type Command-Tab (Browse).**
- ☐ **Click on the door.**

You're back in the first room.

- ☐ **Try clicking on the black door to reenter room 2.**

Stackware Engineering Revisited

Uncovering a Bug

At this point you might notice a minor flaw, or **bug**, in our simulated dungeon: The bed has moved back to its original position. Of course, that's because the Door button was linked to the card with the bed in the corner. If we had anticipated this problem, we could have set it up so that moving the bed blocked the door; then the bed would have to be moved back (via another button) before we could go through the door again.

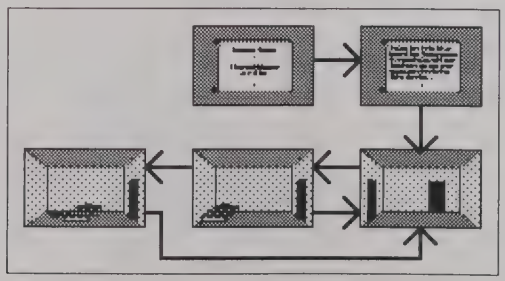
Since this is only a game, we'll sidestep the problem by assuming that either the bed is an enchanted homing bed or this dungeon has incredible room service. And we'll make a note of one of the most important principles of stack design:

> Time spent planning a stack is almost always less than the time wasted repairing an unplanned stack.

Planning a Network

A late plan is better than no plan at all, so let's diagram what we've done so far and figure out where we're going from here.

Here's our basic map:



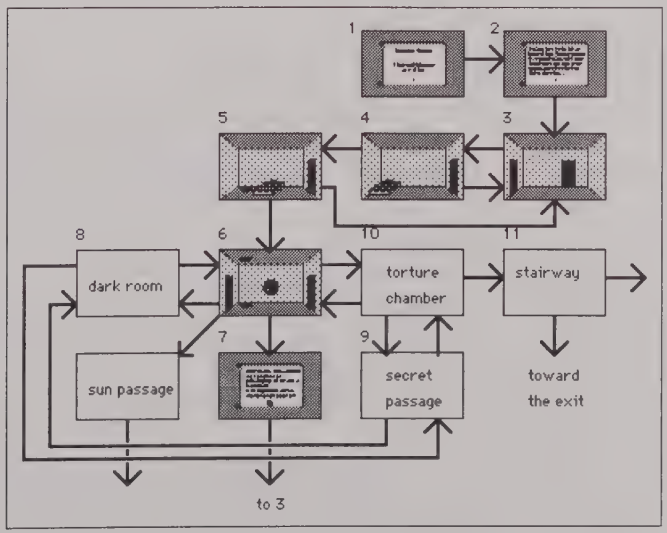
Map Tips

- When you're creating a complex stack, minimize errors by mapping the stack first.
- Show all possible links between cards on the map with arrows.
- Label the arrows with button names so the connections are obvious.
- Number the cards on the map to represent the order they'll be arranged in the stack.
- When you're constructing the stack, use the map as a visual checklist, checking off cards, links, and sections as you complete them.

One thing is clear from the map: This game, so far, isn't very interesting. We need more passages and pathways. We can start by adding a room with several exits: a couple of doors, a hidden passage, and a pit. Each of these exits will lead to other cards.

In the spirit of a good maze, some of these paths should be dead ends. We can, for example, have the Pit button link to a "Sorry, but you just fell into a pit of starving alligators" card. Other paths should wind back into rooms we've already seen. The two doors in the current room might connect to each other via a secret passage. Still other buttons will branch into multiple paths, only one of which ultimately leads to freedom.

We don't have time to develop a complete, lose-yourself-for-an-afternoon maze in this session, but our next diagram might look like this:



This kind of free-form structure is called a **network**. Network structures tend to be more error prone than the sequences and loops we've worked with so far, because there are no real rules about the way they should be put together, and because there are so many loose ends to take care of. It's easy to leave things unconnected or accidentally connect them incorrectly. That's why a map is so important.

Remembering the User

When we discussed software engineering in the last session, we focused on the user interface, developing guidelines to help users find their way around in a stack. Have we applied those guidelines this time around? We've provided the user with no Home or Quit buttons on any of the maze cards, and we've hidden all of the buttons; this stack is almost user-hostile by our earlier standards.

Of course, that's the point. If the buttons were all labeled "this way out" and "dead end" it wouldn't be much of a game. Still, a large, well-designed adventure game allows the player to quit before finishing, often allowing the player to save the current position so he or she can return at a later time. We're not equipped to do that in this session, but we can generalize the idea to an important principle that will make all of our stacks better:

> Always look at your stack design from the user's point of view.

Secondhand Art

Since we're using the same backgrounds for so many rooms, it's important to add embellishments to each room so that it doesn't look like the last one. Embellishing dungeons can be time consuming, especially for nonartists. But we can always borrow.

Copying a Card

Let's start by borrowing a whole card; the one we just created.

- ☐ **Select Copy Card from the Edit menu.**

A copy of this card—graphics, buttons, and all—is on the Clipboard, waiting to be pasted. We'll paste it as soon as we create a new button for the secret passage, using *almost* the same procedure we've done many times.

- ☐ **Select the Button tool.**
- ☐ **Select New Button from the Objects menu.**
- ☐ **Drag and stretch the new button so it covers the hole.**
- ☐ **Double-click on the button to open the Button Info dialog box.**
- ☐ **Click on transparent.**
- ☐ **Click on Show name to turn off that option.**
- ☐ **Type "hole under bed" (optional).**
- ☐ **Click on LinkTo....**

Here's the new twist. Instead of using the New Card command, we'll paste a new card from the Clipboard.

- ☐ **Type Command-V (Paste Card).**
- ☐ **Click on the This Card button.**

As usual, you're carried back where you came from, but this time it's hard to tell, because the two cards look identical—for now.

- ☐ **Type Command-3 (Go Next).**

Deleting and Rearranging Graphics

It's easy to forget where we are, these two cards look so much alike. Let's change that.

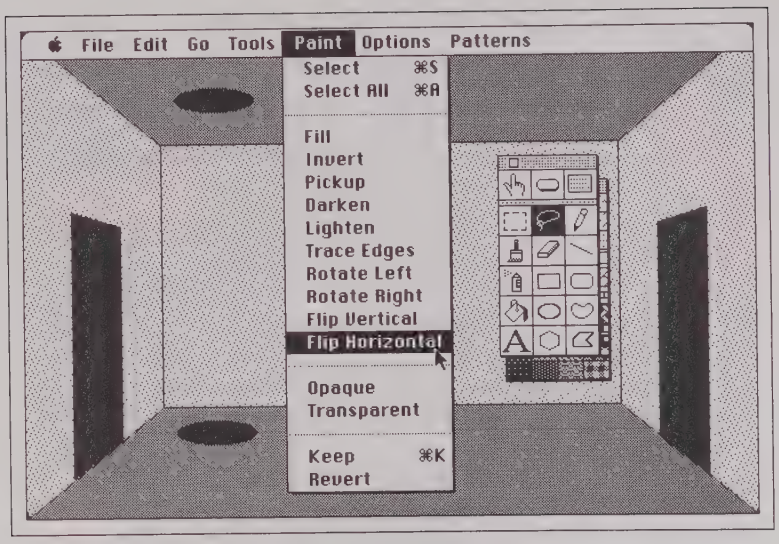
- ☐ **Use the Lasso to select the bed.**
- ☐ **Press the Delete (Backspace) key to delete the selected bed.**
- ☐ **Select the black hole and Shift-Option-drag it straight up to the ceiling, leaving a copy behind.**

The top hole is the bottom of our secret passage; the bottom hole is a pit.

- ☐ **With the Option key held down, select the door with the Selection rectangle.**

This causes the Selection rectangle to shrink around the door like the Lasso.

- ☐ **Option-drag it to make a copy while leaving the original.**
- ☐ **Select Flip Horizontal and position the new door on the left wall.**



Borrowing Clip Art

Now let's add some professionally drawn art.

- ☐ **Type Command-H to go Home.**
- ☐ **Click on the Art Ideas icon.**

Art Ideas

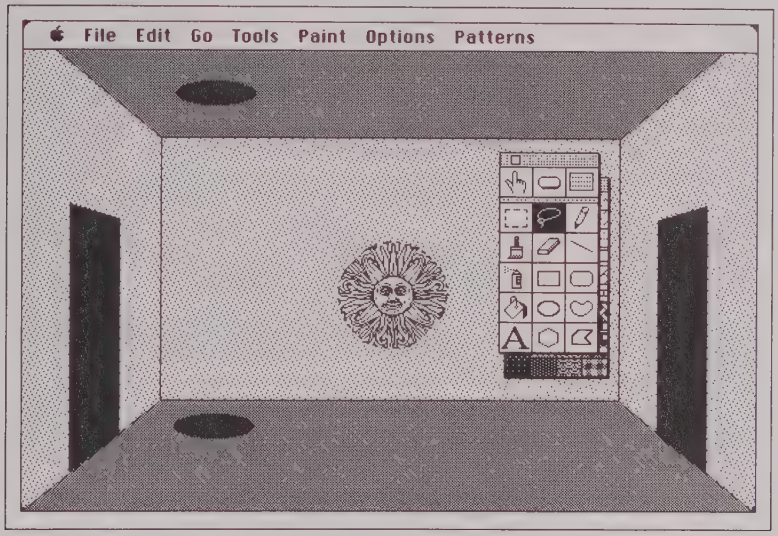
- ☐ **Click on the Science, Astronomy, Chem... icon on the second page of the index.**

- ☐ **Lasso (or Option-select with the Selection rectangle) one of the sun carvings.**

- ☐ **Type Command-C (Copy Picture).**

- ☐ **Type Command-~ (Go Back) four times to return to the room card.**
- ☐ **Type Command-V (Paste Picture).**
- ☐ **Position the carving in the center of the wall.**

Here's the decorated room:



Using Clip Art

Clip art is available in a variety of formats. The Art Ideas stack contains HyperCard clip art, but there's a much wider variety of HyperCard-appropriate clip art stored in **MacPaint** files. The **Import Paint...** command allows you to directly import MacPaint pictures into HyperCard, but this command gives you no control over what part of the 8 x 10 inch MacPaint image you'll use.

It's often more effective to open the clip art document with MacPaint, Studio 1, or some other paint application, copy the portion of the picture you want to the Clipboard, return to your HyperCard stack, and paste. (If you have several images to import, you can paste them all into the **Scrapbook desk accessory** to minimize trips between your paint program and HyperCard.)

Appendix A describes paint applications, clip art sources, and **scanners** that allow you to turn drawings and photographs into HyperCard art.

Borrowing an Old Background

It's traditional in this kind of adventure game to display a message of condolences if the player blunders into an untimely death. We need to display just such a message if our unsuspecting player clicks on the pit. We should display the message on the scroll background that we used for our opening cards. How can we do that, though? If we create a new card here, the card will have the room background. If we go back to card 1 and create a new card, the background will be correct but the card will be in the wrong place. The solution is simple: Go to card 1, create a new card, and move that card to the end of the stack where it belongs.

- ☐ **Type Command-1 (Go First).**
- ☐ **Select New Card from the Edit menu.**

You're looking at a new card with a scroll background. It's located right after the first card in the stack.

- ☐ **Select Cut Card from the Edit menu.**

The new card is now on the Clipboard.

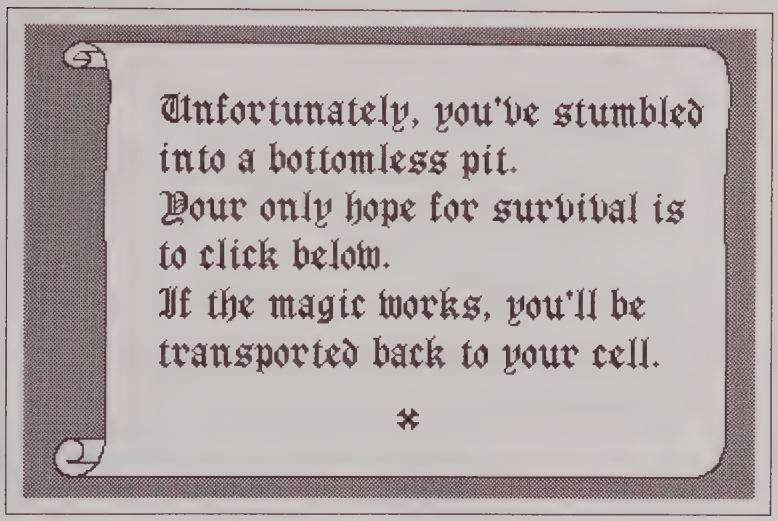
- ☐ **Type Command-4 (Go Last).**
- ☐ **Type Command-V (Paste Card).**

The card is now at the end of the stack.

- ☐ **Type Command-2 to go to the previous card in the stack.**
- ☐ **Select the Button tool.**
- ☐ **Select New Button from the Objects menu.**
- ☐ **Drag and stretch the new button so it covers the hole in the floor.**
- ☐ **Double-click on the button to open the Button Info dialog box.**
- ☐ **Click on transparent.**
- ☐ **Click on Show name to turn off that option.**
- ☐ **Type "pit" (optional).**
- ☐ **Click on LinkTo....**
- ☐ **Type Command-4 (Go Last).**
- ☐ **Click on the This Card button.**
- ☐ **Use the Browse tool to test the button.**

You've now linked the pit to the scroll card. It's time to fill in the scroll.

- ☐ **Double-click on the Paint Text tool, select your traditional font and size, and click OK.**
- ☐ **Type a message on your scroll like this:**


Unfortunately, you've stumbled into a bottomless pit.
Your only hope for survival is to click below.
If the magic works, you'll be transported back to your cell.

There's only one problem. To find out what it is,

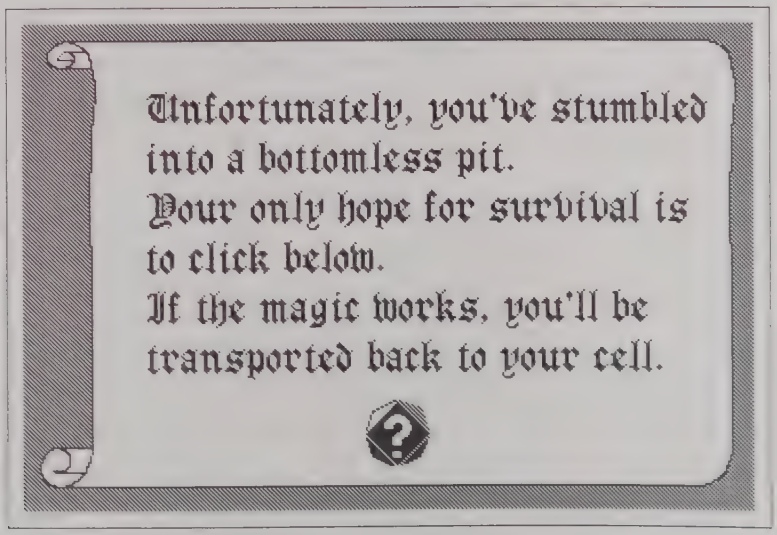
- ☐ **Type Command-Tab (Browse) and click on the button.**

This button, installed in the background, always goes to the next card, which in this case means wrapping around to Card 1. We want a different kind of button on this particular card. If we change the background button's link, that will change the link for all cards with this background. We can get around this problem by taking advantage of basic layering principles.

> Buttons on the card layer can cover background buttons in the same way that card layer graphics cover background graphics. If a card layer button covers a background button, the card layer button takes precedence.

- ☐ **Type Command-4 (Go Last).**
- ☐ **Select the Button tool.**

- ☐ Select New Button.
- ☐ Position and shape the new button so it covers the background button.
- ☐ Double-click on the new button to open the Button Info dialog box.
- ☐ Click on opaque.
- ☐ Turn off Show name.
- ☐ Click on Icon....
- ☐ Select a magical icon and double-click.
- ☐ Double-click on the icon to bring back the Button Info box.
- ☐ Click on LinkTo....
- ☐ Type Command-3 until you see the first room card.
- ☐ Click on This Card.
- ☐ Use the Browse tool (Command-Tab) to try the new button.


Unfortunately, you've stumbled into a bottomless pit.
Your only hope for survival is to click below.
If the magic works, you'll be transported back to your cell.

As your maze expands, you'll probably have lots of dead ends. Each time you need one, you can copy this card, complete with text, buttons, and background. When you paste the copy where it goes, all you need to change is the message. (In the interest of user friendliness, it might be a good idea to add a Home button to this card before you copy it.)

Changing Links

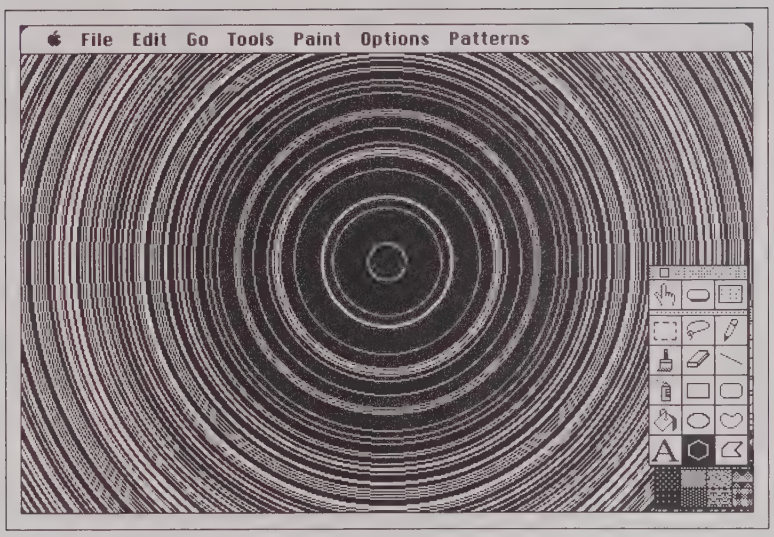
Let's finish by illustrating another important design principle:

> Don't just tell the user; *show* the user.

Just for fun, let's add a graphic representation of the bottomless pit before the closing message. That means we'll have to insert a card between the pit room card and the final scroll, reconnecting the links as we do. The process is simple, provided you pay careful attention to what you're doing so you don't forget to tie everything up properly.

- ☐ **Type Command-4 (Go Last) and Command-2 (Go Previous).**
- ☐ **You should be looking at the room with the entrance to the pit.**
- ☐ **Select the Button tool and double-click on the Pit button.**
- ☐ **Click on LinkTo....**
- ☐ **Select New Background.**
- ☐ **Click on This Card.**
- ☐ **Type Command-3 to go to the new white card.**
- ☐ **Double-click on the Regular Polygon tool to choose a shape.**

- ☐ **Click on the circle.**
- ☐ **Select Draw Centered and Draw Multiple from the Options menu. (If there's a checkmark in front of Draw Filled in the menu, select Draw Filled to turn it off.)**
- ☐ **Drag from the center of the screen outward with an even motion, creating a series of concentric circles as you do. Without lifting the button, drag back toward the center if you need to fill gaps.**



If necessary, double-click on the Eraser and repeat until you get the endless tunnel effect you like. (If you prefer, try one of the other shapes.)

We need a button to make this into a click-anywhere-to-continue screen.

- ☐ **Select the Button tool.**
- ☐ **Create a screen-sized New Button.**
- ☐ **Double-click on the button to open the Button Info dialog box.**
- ☐ **Make it transparent.**
- ☐ **Turn on Auto hilite.**
- ☐ **Turn off Show name.**
- ☐ **Click on LinkTo....**
- ☐ **Type Command-3 (Go Next).**

You should be looking at the "sorry" scroll.

- ☐ **Click on the This Card button.**
- ☐ **Use the Browse tool to test the new button.**

Finishing the Game

There's still much to do before this game is finished, but you have the tools and techniques you need to carry it through on your own. If you want to

continue, add buttons to the two doors and the sun picture in the last room, and link those buttons to new rooms. (You already have one button installed on the right-hand door, but it links to the wrong card. Why?) Don't be afraid to be creative, but keep in mind a few tips as you create:

Keeping Your Stack Orderly

- As you add cards, try to keep the overall stack in some kind of logical order.
- Remember that whenever you create a new card, that card is inserted immediately after the current card. If that's not where you want it, use Cut Card and Paste Card to move it.
- A nonlinear structure has to have some buttons that jump to other parts of the stack, but it's best to keep cards in logical groups as much as possible.
- Flip through all the cards in your stack with Command-3. It's easy to leave unfinished, unwanted cards embedded in a stack, and this is a good way to find them.
- If you find several cards that look alike, use temporary ID marks to flag them so you can tell them apart while you're troubleshooting; you can remove the extra cards and the ID marks when the job is done.
- Use Option-Command to reveal hidden buttons on each card; try each of these buttons. When you copy cards, you often end up with buttons that you didn't want linked to places you don't want to go.

As you expand this stack, practice borrowing cards, art, and buttons from other stacks until you're comfortable with the process. Try the paint tools and menu options that we haven't covered here, referring to the HyperCard Help stack for information and inspiration. Have fun!

Summary

In the process of building a game, you've learned a number of HyperCard concepts that you can apply directly to all kinds of serious and not-so-serious projects. You've used many new painting tools and techniques to create complex graphic scenes. You've put buttons and graphics on backgrounds that were shared by several cards. You've seen the possible interaction between card and background objects, and you've learned how

to take advantage of HyperCard layers to minimize your work and maximize your stack's efficiency and effectiveness. You've also discovered some of the pitfalls of designing and building a complex stack, and learned some principles that can help you to avoid those pitfalls.

In Session 5 you'll learn how to add text fields to your stacks so you can create hypertext documents that transcend the limitations of traditional printed media. The principles you learned in this session will apply directly to the project you'll create then.

Key Words

absolute reference (link)
adventure game
Background command (Command-B)
background layer
bug
card layer
clip art
Copy Card command
Curve tool
Draw Centered command
Draw Multiple command
Edit Pattern... command
FatBits
hand tool (Option-FatBits)
Import Paint... command
interactive fiction
logical reference (link)
MacPaint
network structure
New Background command
Opaque command
opaque graphics
oval tool
Paste Card command (Command-V)
Regular Polygon tool
scanner
Scrapbook desk accessory
simulation
stretching/shrinking graphics (Command-drag)
stripes (on menu bar)
Transparent command
transparent graphics

Self-Testing Exercises

1. List four reasons why it's a good idea to include common elements of cards in a shared background.
2. Is it possible to have a background button in a two-card, one-background stack, that takes you back and forth between the two cards? Explain.
3. What do each of these HyperCard tool symbols represent?

   a.

   b.

   c.

4. Describe three different ways you can select an irregularly shaped object on the current layer of a card.

5. How can you switch to FatBits without using the menu? How can you get back to "skinny bits"?

6. Describe the difference between logical reference and absolute reference to another card. Under what circumstances would each of these be most appropriate?

Projects

1. Create a map of your city with a list of points of interest along the side or at the bottom. Make each one of those points of interest into a button that, when clicked, highlights the named point on the map. Put the map in the background and create one highlighted card for each item. Should the buttons be on the card layer or the background layer? Why? Variations: Do the same thing for your house, your state, the world, or whatever you can think of.

2. Make a line drawing of a familiar object on a HyperCard card. Select that object with the Selection rectangle and capture the screen image as a MacPaint file with the Command-Shift-3 keystroke combination. (The Mac will save the image as a file named Screen 0, Screen 1, . . . , or Screen 9.) Import that captured screen back into HyperCard and use it to create a demonstration of these effects in HyperCard's Paint menu: Fill, Invert, Darken, Lighten, Trace Edges, Rotate Left, Rotate Right, Flip Vertical, and Flip Horizontal. Your screen should list these commands down the left side in Chicago font. Clicking on any of the command names should change the screen so that (a) the command name is highlighted (inverted); (b) the selected part of the picture is transformed just as the command would transform it; and (c) an Undo button appears to take the user back to the previous card. Think carefully about what parts of the picture should be in the background before constructing this stack.

SESSION 5

Objectives

By the end of this session you should be able to

- Create a hypertext document with dynamic links between text fields
- Create and modify different kinds of HyperCard text fields
- Design a complex document with a linear structure, a tree structure, and a network structure
- Understand how different document structures can be used by different kinds of hypertext readers
- Apply the principles of software design and stepwise refinement to the process of building a HyperCard stack

5

CREATING HYPERTEXT:

NONLINEAR WRITING

The Problem

Your CS 101 instructor has asked you to produce a paper on basic computer anatomy. He wants you to write it in such a way that readers can either read it as a tutorial or use it as a reference to look up particular terms. He expects you to produce a document that presents an overview of the material but that allows the student to dig deeper into any concept to learn more. He also wants the paper to work as a reference guide for quickly looking up computer hardware terminology. This kind of flexibility doesn't seem possible in a traditional term paper. Perhaps HyperCard can help.

Introduction

HyperCard is often described as Apple's brand of hypertext. But as you've seen in the last two sessions, HyperCard is, at its core, a kind of hypergraphics tool. Nonetheless, text manipulation is an integral part of HyperCard, and that's what we'll focus on during this session. In order to create a hypertext document that gives the user maximum flexibility, we'll design our document from the top down, filling in details as we go.

Designing Hypertext

Defining the Problem

When all is said and done, a computer is a tool for solving human problems. If we keep that in mind when we're working with HyperCard, it's easy to remember and apply one of the most important rules of thumb in programming:

> *Start by defining the problem clearly.* A program (or stack) with stunning graphics, elaborate special effects, impressive content, and mindboggling structure is worth less than a simple stack that addresses and solves the problem at hand.

Our problem, defined at the beginning of this session, is to create a document that can be used by beginning CS 101 students to gain an understanding of how the basic parts of a computer function. Our readers, after a brief introduction, should be able to read the material sequentially, like a book, or explore particular topics in detail while ignoring others, or look up words for quick definitions and illustrations. We need a hypertext document.

(A word on vocabulary: **Hypertext**, as it was defined by its original champions, deals mainly with text documents. Today, many hypertext systems include graphics as well as text. The creators of HyperCard insist that it is not a hyper*text* system, but a **hypermedia** system, because it allows the linking of cards containing text, graphics, and sound. Until the world agrees on a universal hyper-vocabulary, choose your favorite term.)

The implicit assumption in our statement of the problem is that we are to use appropriate resources—in this case, HyperCard. (This may seem obvious, but it's still important. We could not use HyperCard if there were no Macintosh computers available to the potential users.)

Breaking the Problem into Pieces

The problem definition may seem clear, but it is vague in its generality. Computers don't tolerate vague, so we need to break the problem into more specific pieces.

> After you've defined the problem (what you're trying to do), break it down into subproblems (how you're going to do it).

What, then, are the subproblems? If we were writing a traditional term paper, we might divide it into (1) introduction, (2) body, and (3) conclusion. We've agreed to throw out the traditional model, but before we do, let's examine each traditional component critically to see if it has a place in our hypertext document.

The introduction usually prepares the reader by (a) stating the goal(s) of the paper and (b) providing the necessary orientation for understanding the material in the paper. A hypertext reader needs to be prepared, too. A potential reader is not likely to show much interest in browsing through your stack if it's impossible to figure out what he or she is going to get out of it. Furthermore, since a hypertext document can be read in a variety of ways, it's especially important to start by making sure the reader is primed with whatever basic information is needed to make use of the stack: underlying assumptions, layout of the stack, and so on. So our introduction might contain a title card, how-to-navigate-this-stack information, and a brief introduction of the concepts that will apply throughout the stack.

The body of a traditional term paper presents arguments, ideas, facts, quotations, and sources, tied together in prose that makes sense when read from beginning to end. In this case, we'll be presenting facts and ideas, but we can't assume that each reader will be plowing through the material sequentially. For some readers, the "body" of our hypertext document might be more like an encyclopedia than a term paper. So we need to make sure that each "chunk" makes sense whether the reader approaches it from point A or point B. If the structure is complex, it's safest to make each chunk self-contained, assuming only that the reader has read the introduction.

The conclusion of a term paper generally presents a summary of the main idea(s) so that the reader comes away seeing the forest as a whole after examining all these trees in the body. In many hypertext documents, there's no real conclusion, because there's no way to make sure the self-guiding reader ends the journey on a particular card. If our stack is strictly a reference stack (like a dictionary), we probably don't need to worry about a conclusion. But if we want the stack to work as an introductory

tutorial, too, we may want to guide readers through a closing card or two on their way home. We can defer that decision until we have a clearer idea of our structure.

Let's use these ideas to break the project into pieces:

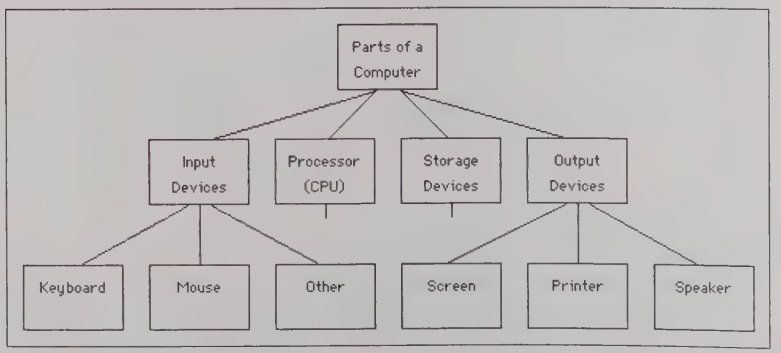
1. Introduction (title screen, background/overview, navigation instructions)
2. Body (details of computer components upon request)
3. Conclusion (if necessary and appropriate)

The components of this structure are still too vague for a computer to deal with, so we need to practice what's known in the industry as **stepwise refinement**. Stepwise refinement, like building the outline for a term paper, involves repeatedly breaking each component into smaller and smaller pieces until each piece is small enough to deal with easily.

Fleshing Out the Body

Let's start by considering the body, since that's the most important part of the stack and it will determine what the reader needs to be told in the introduction. The reader or user should be able to look at some kind of menu screen that shows the basic components of a computer and to click on each component to get more information about it. For example, the user might click on Output to see a general description of output devices and a an array of buttons to click for more information about specific output devices such as screen, printer, and speaker.

Here's a partial map of what this part of the stack might look like:



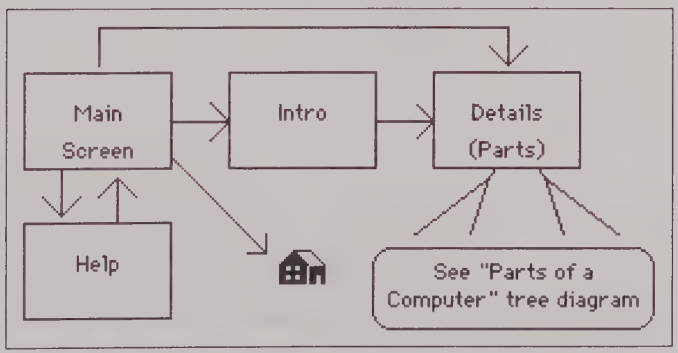
This is called a **tree structure**. (In computer science, trees are generally drawn with **roots** at the top and **branches** that grow downward.) Unlike the network structure we used to create our dungeon, a tree structure has

well-defined characteristics that make it predictable and easy to navigate. The advantage of a tree is that we can add more leaves to allow for more detailed descriptions or delineations. The user simply has to click on buttons to go deeper and deeper into the hierarchy and expose more details. Of course, we should provide a button on every card for returning to higher levels of the hierarchy, another for returning to the main menu page, and (possibly) a third for taking the user to a map that shows an overview of the stack. Since these buttons should appear on every card, we should consider putting them in the background.

Refining the Introduction and Conclusion

The main title page should allow the user to choose a path through the information by clicking on one of several buttons. New users, in addition to needing some kind of how-to-use-this-stack information, may need an introduction to make sure they have the background needed to understand the material in the stack. They're also likely to want to read everything in sequential order to make sure they don't miss anything. Returning customers may want to bypass the introduction and dive right into the details of the main tree. Those users who are really in a hurry might want to just use the Find command to jump directly to a particular subject. So our opening screen can be thought of as a main menu with several options: Intro (background information), Details (the tree-structured body), Find (quick search), Help (stack instructions), and Home (exit).

Here is a possible map of the introduction (without the Find option, which is difficult to show in this kind of diagram):



We might want to include a conclusion later, but for now our major concern is that the user have a way to exit the stack and a way to get to another part of the stack from anyplace in the stack. We can combine these two functions if we assume that the main menu card will have a Home button, so that one exit to the main menu will allow the user to exit to Home

or to explore another part of the stack. Instead of a conclusion, we can include a simple Exit button that will return the user to the main menu card.

Outlining the Stack

An outline of our stack as we have defined it so far would look like this:

1. Introduction: main screen with title, menu, buttons
    a. Help button that displays instructions, returns to main menu screen
    b. Intro button that displays introductory material, then leads to main body
    c. Details button that bypasses intro, goes to Body (tree)
    d. Find button displays Find command in message box
    e. Home button to exit
2. Details (parts of a computer)
    a. Buttons for four computer components (input, CPU, storage, output) and an Exit button
    b. For each component, define component type, provide buttons offering more details (such as description of keyboard), and provide an Exit button

Analyzing the Structure

With this outline as the structure, how do we decide what goes in the background?

What Goes in the Background?

- Which elements repeat themselves in your outline or map?
- Which elements could appear on most or all cards without causing unnecessary confusion or complication?

Any element that appears in either answer should be considered as a candidate for the background(s).

If we apply these principles to our outline, we see that the Exit button appears on every card in the details section, so we'll definitely want to include it in the background. Does this mean that the details section should

have a different background than the introduction? It could, but before we decide that, let's look at the second question. Virtually every button we've listed can appear on multiple cards. In fact, our stack is more versatile if the user can backtrack to the introduction from anywhere in the stack.

Designing the User Interface

There's always a risk that too many buttons will overwhelm the user unless they're arranged in a clear, understandable way. Since we're using several buttons on every card, we should label the buttons rather than depending on icons to make their meanings clear. We'll also arrange the screen so that the universal buttons are all together in a row, and so that they all look and act in the same way. For example, in this stack we'll arrange the buttons across the top of the background, and we'll make them all rounded-rectangle buttons that auto-hilite when clicked. These kinds of details might seem unimportant, but they can make a tremendous difference to the people who will be using your creation.

> Design your stack so that the buttons have a consistent look and feel, so the user knows what to expect when clicking them.

One way to determine what kinds of user interfaces work best is to look at stacks that work well and study their user interfaces. In fact, it's sometimes helpful to borrow directly from these stacks as we're designing our own. That's what we're going to do this time: We'll borrow a background from the Art Ideas stack and modify it to meet our specific needs. (We'll also be borrowing from the Button Ideas stack and the Phone stack; make sure you have all of these stacks on your file server, hard disk, or disks when you do this project.)

At this point in the planning process, we should draw several prototype screens to experiment with different user interfaces and designs. We might find that some of our ideas don't work well when actually applied to paper or pixels, and have to come up with alternative approaches.

> Never be afraid to backtrack when designing a stack. It's almost never worthwhile to continue building on a flawed plan.

There are many more design decisions to be made, but let's use our fast-forward button to jump to the moment when we've completed the design

and we're ready to actually build the stack. From that perspective, we can look back on those decisions that we bypassed.

Making a Master Menu

Creating a Working Stack

- ☐ **Turn on your computer, boot it up, and open your Home stack.**

Art Ideas

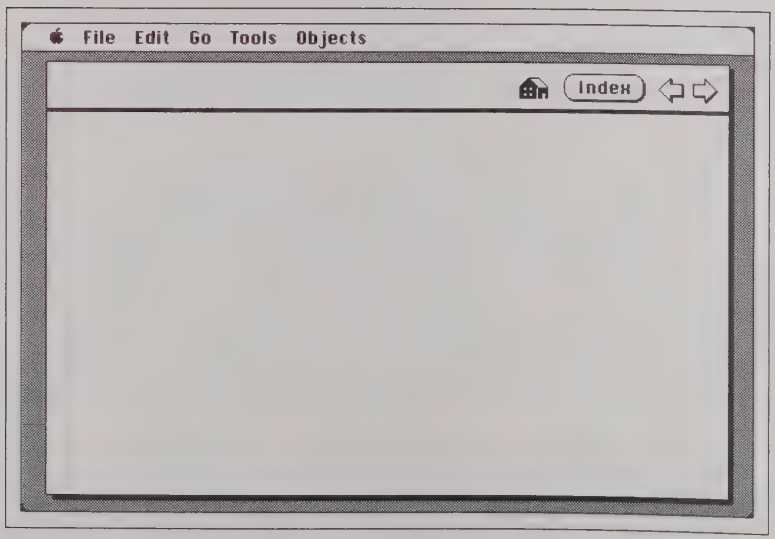
- ☐ **Click on the Art Ideas icon and locate the card in Art Ideas called Macintosh, Apple....**

We're going to use some of these predrawn images to illustrate our stack. But we're also going to borrow the background from this card and use it as raw material for our background.

- ☐ **Select Copy Card from the Edit menu.**

A copy of the entire card is now on the Clipboard.

- ☐ **Select New Stack from the File menu.**
- ☐ **Name the stack "Computer Anatomy," locate the appropriate disk and folder in the dialog box, make sure Copy current background is clicked on, and click New.**



We have a new stack with only a background. But we have a card on the Clipboard.

- ☐ **Type Command-V (Paste Card).**

Eventually we'll delete this card, but it's handy to have it tacked onto the end of the stack where we can easily find it when we need it.

Creating an Exit Button

The background that we copied for this stack already has four buttons: Home, Index, and the left- and right-arrow buttons. We know what three of those buttons do, but what does the Index button do? In the Art Ideas stack, the Index button takes you to the first card in the stack—the Index card. Recall from the last session that this could be accomplished with an absolute reference (Go to card Index, wherever it is) or a logical reference (Go to the first card in the stack, whatever it might be). If this button uses a logical reference, we can use it as our exit to the main (first) card of *this* stack.

- ☐ **Click on the Index button.**

Since we've been transported to our blank first card, we know that this is the button we need—provided we change the name of the button.

- ☐ **Drag the Tools palette onto the screen and select the Button tool.**
- ☐ **Double-click on the Index button, type "Exit" to change its name, click Auto hilite, and click OK.**

This Exit button should appear on every card *except the first one*; its function will be to take the user to the first card, where the Home button allows the user to exit from the stack. The Home button, however, should appear *only* on the first card. We can take care of both of these needs by pasting the Home button on the card layer of the first card right on top of the Exit button so that only the Home button is visible (and usable).

> A button on the card layer takes precedence over a button in the same relative location on the background layer.

- ☐ **Select the Home button and type Command-X (Cut Button).**

(You don't need to be working in the background to select a background button that's not covered by another button.)

- ☐ **Type Command-V (Paste Button).**

While this causes no visible change, it has, in fact, had the desired effect: The button has been cut from the background and pasted onto the card layer.

- ☐ **Drag the Home button so that it sits in the center of the Exit button.**

This isn't quite right, because the Exit button is still showing through.

- ☐ **Select the Selection rectangle paint tool (the marquee) and drag a rectangle around the Exit button, being careful not to touch the dark lines that cross the screen above and below the button.**

- ☐ **Select Opaque from the Paint menu.**

The card layer is no longer transparent in that area, so the background Exit button is hidden.

Borrowing a Find Button

Our background button bar also needs a Find button. Since we haven't covered script writing for buttons, we'll have to find a button that does what we want and copy it into our stack. The **Button Ideas stack** is a collection of buttons that are pre-programmed to perform common tasks; it has just such a button.

- ☐ **Type Command-H (Go Home) and click on the Button Ideas icon.**
- ☐ **Click on the Text button to select the Text button card.**

- ☐ **Select the Button tool, click on the Find button, and Type Command-C (Copy Button).**
- ☐ **Press the Go Back key (Esc or ~) until you're back to the first card of your stack.**
- ☐ **Type Command-B to go to the background.**
- ☐ **Type Command-V (Paste Button).**

- ☐ **Drag the Find button on top of the Exit button and shrink it so that both buttons are the same size; then drag the Find button so it's to the left of the Exit button.**

When you click on this button with the Browse tool, the message box should appear with the Find command already typed, just as if you had selected Find from the menu.

Creating Placeholder Buttons

We still need buttons to take the user to the introduction and to the body of the stack. These can be simple linking buttons, once we have something to link them to. For now, let's simply create visual placeholder buttons and leave them unlinked.

- ☐ **Select New Button from the Objects menu.**
- ☐ **Double-click on the new button to bring up the Button Info dialog box.**
- ☐ **Click on Auto hilite, type "Details," and click OK (or press Return).**
- ☐ **Make this button the same size as the other two and position it to the left of the other buttons in the button bar.**
- ☐ **Repeat the process to create another button named Intro and put it in the menu bar.**

Your background button bar should look like this:

Borrowing a Help Button

In the card layer of the first card, we need one more button: a Help button for first-time users. We could create one in the usual way and link it to a Help card, but let's borrow a button that does something fancy instead.

Phone

- ☐ **Go Home (Command-H) and click on the Phone icon.**
- ☐ **In the Phone stack, click on the Question Mark button. Click again on the field that appears.**

This button does not transport you to a different card; it simply makes a field called info pop up on top of the current card. We can create the same effect in our stack by cloning the button to our first card and creating a field there with the same name.

- ☐ **Select the Button tool, click on the ? button, and type Command-C (Copy Button).**
- ☐ **Press the Go Back key until you're looking at the first card of your stack.**
- ☐ **Type Command-V to paste this button onto the card layer.**
- ☐ **Position the button in the button bar to the left of the others.**

- ☐ **Double-click on the ? button, click Auto hilite, and click OK.**

This button is useless until we create an info field; we'll do that as soon as we add the title to our card.

Field Day

Typing a Title

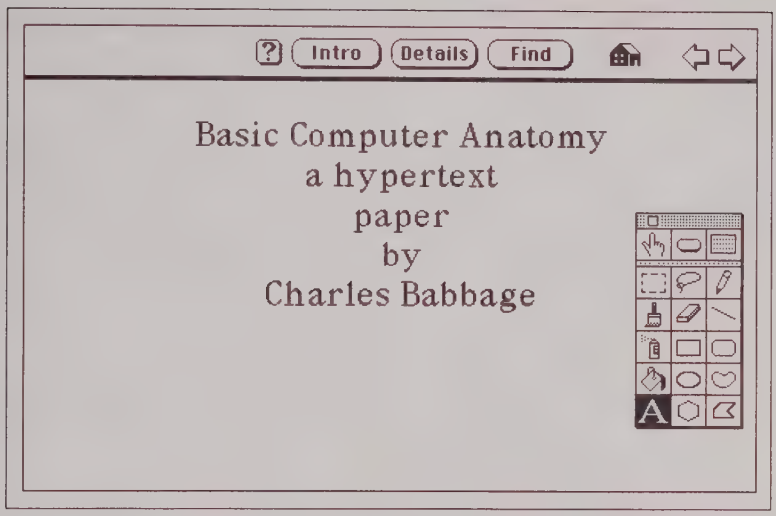
Before we start playing with fields, let's create a title for our main screen using familiar paint text.

- ☐ **Double-click on the Paint Text tool, choose a font that looks good in a large size (as usual, that depends on what's in your System, but Geneva is generally available in large sizes), select the largest available size, click Align Center, and click OK.**
- ☐ **Click below your button bar in the center of the screen and type a title and credit line, as shown at the top of the next page.**

This paint text will dry as soon as you click on any other tool. Now we'll start working with field text, which is a different kind of beast altogether.

Making a Heading Field

- ☐ **Click on the Field tool in the upper right corner of the Tools palette.**



You should see the outline of a background field hidden behind the row of buttons at the top of the card. This field came as a bonus with the background that we borrowed from the Art Ideas stack. In that stack, the field is used to hold the title of the card, so that you can use the Find command to search for key words in the title. We can use it for the same purpose in our stack, but not if it's hidden behind all those buttons.

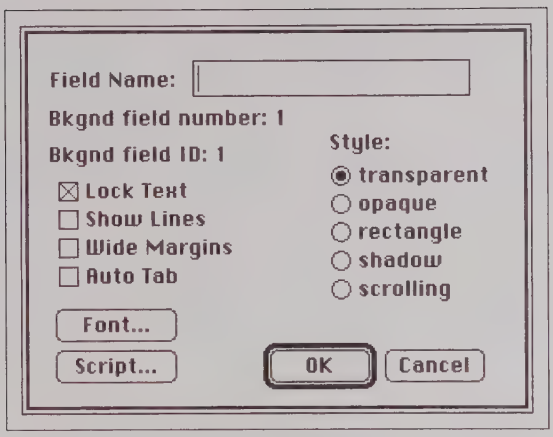
The Field tool allows you to move, resize, and control the characteristics of fields in the same way that the Button tool allows you to work with buttons.

- ☐ **Place your pointer on the left edge of the field and stretch the field so that it extends to the left of the card.**

Fields, like buttons, have Info boxes.

- ☐ **Double-click on the field or select Field Info... from the Objects menu.**

Notice that the **Lock Text** option is checked in this dialog box. Since this was originally a title field, that's appropriate. But we have to temporarily unlock it so we can add titles to it.



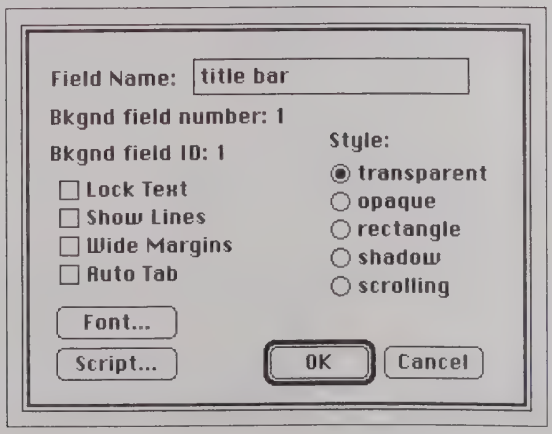
The contents of a field cannot be changed as long as the Lock Text option is checked.

- **Click on the Lock Text box to unlock this field.**

Later, when our hypertext stack is completed, we can click Lock Text again, so that users can't delete or change the title of any cards while browsing.

- **Type "title bar" to name the field.**

This step is optional, but it's a good habit to establish.



- ☐ **Click OK.**

Let's test our new field.

- ☐ **Type Command-Tab, the shortcut for selecting the Browse tool.**

The field should be invisible, because it's a transparent field. The Browse tool should change to an I-beam over the field. For each card in the stack, we'll fill this field with a title that identifies or summarizes the contents of the card. But since the first card already has a title, we can use the field to give the user the most basic instruction.

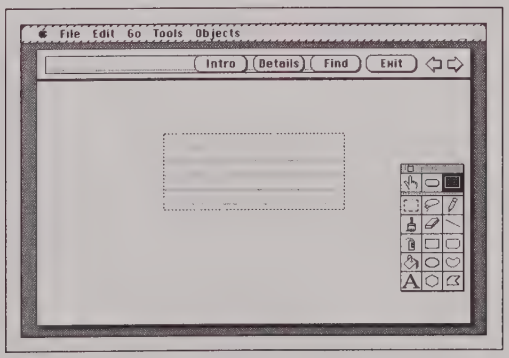
- ☐ **Make sure you're in the card layer; then move the pointer over the field so it turns into an I-beam.**
- ☐ **Click in the field and type:**

Creating a New Field

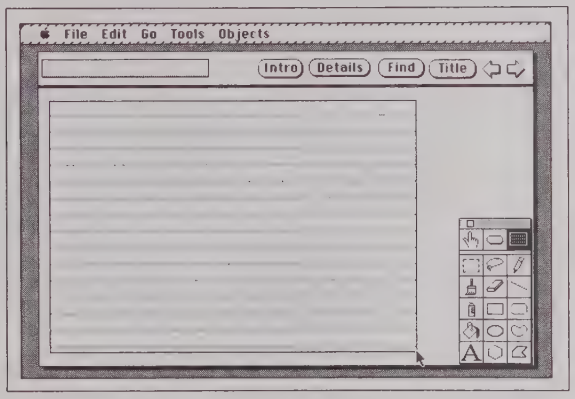
Now we need to add a larger field to the background for storing the bulk of the text.

- ☐ **Type Command-B to go to the Background.**
- ☐ **Select New Field from the Objects menu.**

You'll see a field like this one:



- [ ] **Drag and stretch the field so it fills most of the rest of the card, leaving space on the right for illustrations, like this:**

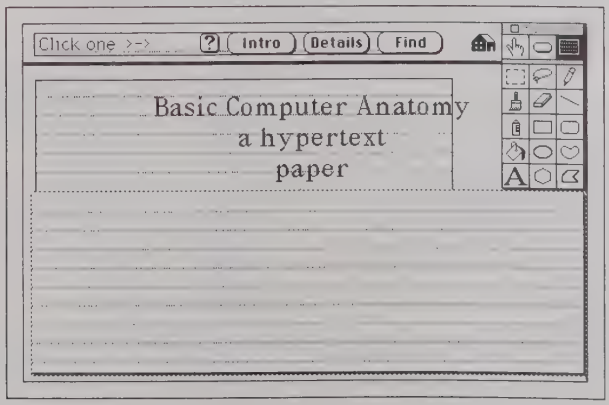


The default text style for a field is 12-point Geneva, which is just fine for our purposes, so this field is ready to use.

Creating a Pop-up Field

We still need to create a card-layer info field to go with the ? button.

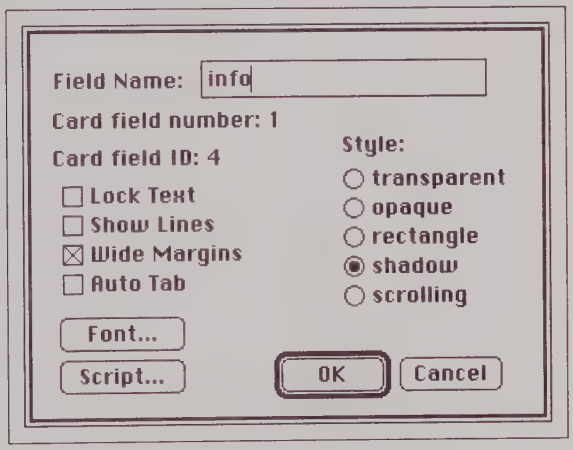
- [ ] **Type Command-B to leave the background and return to the card layer.**
- [ ] **Select New Field from the Objects menu.**
- [ ] **Drag and stretch the field so that it covers the bottom half of the screen, like this:**



- ☐ **Double-click on the field to reveal the Field Info dialog box.**
- ☐ **Type "info" in the field name box.**

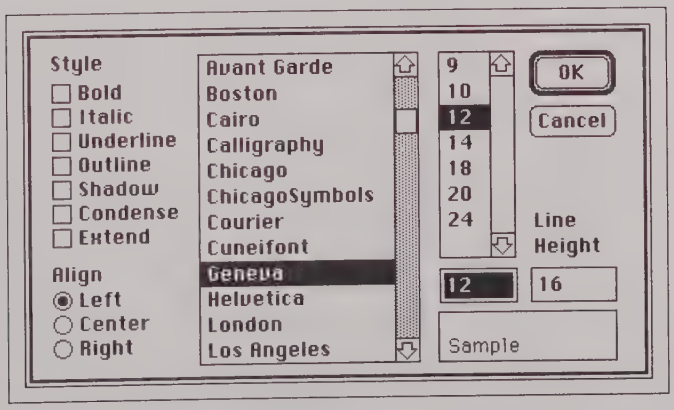
This time the name is mandatory, because the ? button looks for a field named info when it's clicked.

- ☐ **Click on the shadow style button and the Wide Margins check box.**



- ☐ **Click on Font....**

The familiar Text Styles dialog box appears.



- ☐ **Select Geneva, 12, Align Left, and click OK.**

Fonts in the Field: A Cautionary Note

Your choice of fonts for field text should be based on more than just aesthetics. While paint text blends into the pixel pattern on the card or background after you create it, field text remains text forever. Field text is displayed on the screen using fonts stored in the System file, which may or may not be the same fonts that were in the System that was used to create the stack. If the fonts that were used to create the text are not available on the user's System, the System substitutes another font. The results may be aesthetically unsatisfactory or worse. In some cases, substitute fonts may cause text to disappear from the screen because it won't fit in the field.

If you're building a stack that is to be displayed on different computers, you can avoid these potential problems by sticking to the **standard fonts** that are installed in all Macintosh System files. The only fonts that are required in every Mac System are Geneva 9 and 12, Monaco 9 and 12, and Chicago 12. But it's usually safe to assume that several other sizes of Geneva are installed. Times, Helvetica, and New York are commonly used, but not universally available.

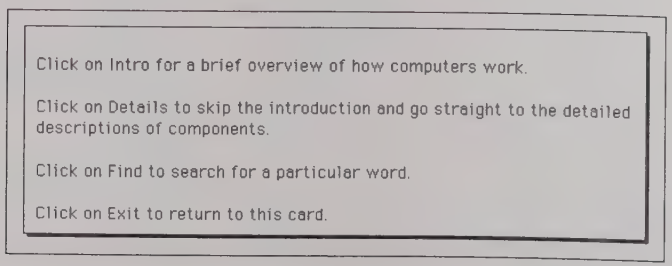
If you must include something in an unusual font and you can't predict what computers will run your stack, (1) use paint text to do the typing, or (2) install the font (provided it's in the public domain) in the stack using the **Font/DA Mover**. (The second option is not recommended for beginners.)

Let's test the field.

- ☐ **Type Command-Tab (Browse).**
- ☐ **Click on the ? button repeatedly.**

The field should appear/disappear each time you click.
Now you need to fill the field with a brief set of instructions.

- ☐ **Click in the upper-right corner of the field to position the insertion bar and type this:**


Click on Intro for a brief overview of how computers work.
Click on Details to skip the introduction and go straight to the detailed descriptions of components.
Click on Find to search for a particular word.
Click on Exit to return to this card.

We don't want the user to be able to change this, so let's lock the text.

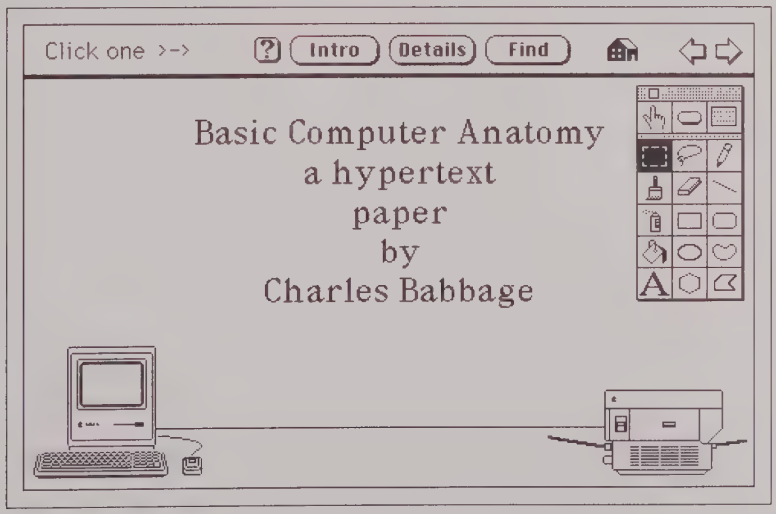
- **Select the Field tool and double-click on the info field.**
- **In the Field Info dialog box, click Lock Text, and click OK.**
- **Click on the Browse tool.**

When you move the Browse tool over the field, it no longer changes to an I-beam; typing is not allowed.

- **Click on the ? icon to turn this pop-up field off.**

When the help (info) field is turned off, the conspicuously white bottom of the card is revealed. That's a good place for a picture.

- **Copy pictures of a Macintosh and a LaserWriter from the last card in your stack (the borrowed card from Art Ideas) and paste them onto this card, moving each item to a lower corner of the card immediately after pasting. Network them with the Straight Line tool, like this:**



Introduction: One Page at a Time

The first card is almost done. We have a title, artwork, and three working buttons: Home, Find, and ?. We still need to provide the Intro and Details buttons with someplace to go. We planned the introduction to be a **linear**

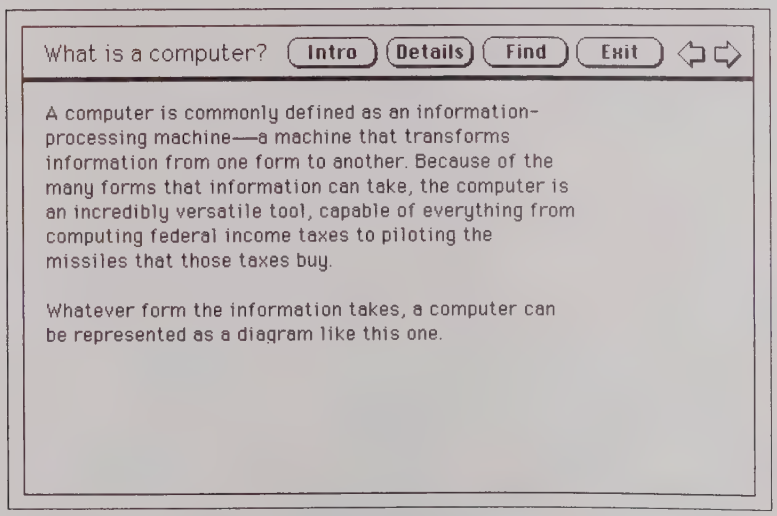
structure—a simple sequence of screens introducing basic computer concepts. Let's create that now.

Create the Introduction

- ☐ **Select the Button tool, double-click on the Intro button, and click on LinkTo....**
- ☐ **Type Command-N (New Card) and click on the This Card button.**

As usual, the new card shares a background with the previous card—including the background's buttons and fields. You can't see the fields, but your browsing pointer should turn into an I-beam whenever you pass over them.

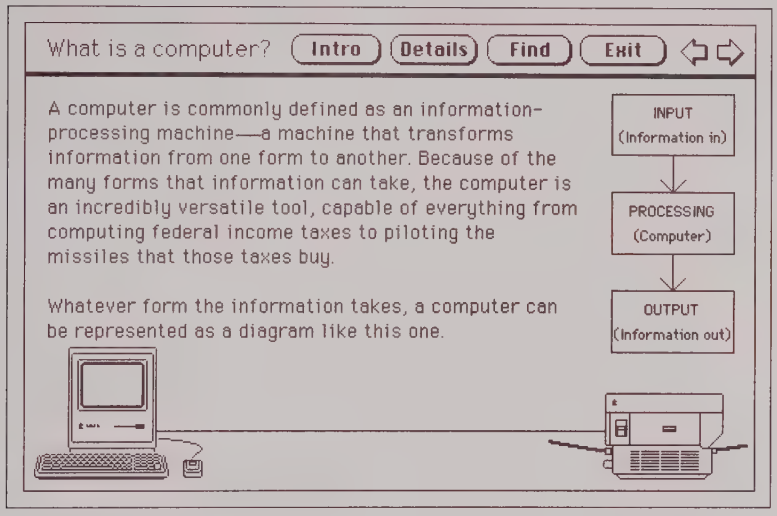
- ☐ **Locate each of those fields and fill them with the following text (for field text, use the Browse tool, *not* the Paint Text tool):**


What is a computer?
Intro
Details
Find
Exit
A computer is commonly defined as an information-processing machine—a machine that transforms information from one form to another. Because of the many forms that information can take, the computer is an incredibly versatile tool, capable of everything from computing federal income taxes to piloting the missiles that those taxes buy.
Whatever form the information takes, a computer can be represented as a diagram like this one.

Let's add some graphics to this card.

- ☐ **Using the left-arrow button in the button bar, return to the first card.**
- ☐ **Select (with the Selection tool or the Lasso) the computer/printer network and copy it to the Clipboard.**
- ☐ **Go back to the second card and paste the selected picture.**

☐ **Use the Rectangle tool, the Straight Line tool, the Clipboard, and the Paint Text tool to create the flowchart on the right side of the card like this:**


What is a computer?
Intro
Details
Find
Exit
A computer is commonly defined as an information-processing machine—a machine that transforms information from one form to another. Because of the many forms that information can take, the computer is an incredibly versatile tool, capable of everything from computing federal income taxes to piloting the missiles that those taxes buy.
Whatever form the information takes, a computer can be represented as a diagram like this one.
INPUT
(Information in)
PROCESSING
(Computer)
OUTPUT
(Information out)

The process for creating the next four cards is similar:

1. Create a new card.
2. Type the appropriate text into each of the two fields (remembering to use the Browse tool).
3. Copy and paste any graphics that are repeated from previous cards (the Option key with the Selection rectangle works well for many of these), and paint any new graphics.

There's no need to explicitly link any of these cards to buttons, because the right-arrow button will automatically take the reader to the next card, whatever it might be.

☐ **Following the steps outlined above, create each of the following cards.**

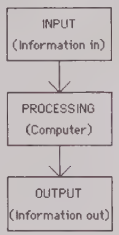
If you're in a hurry, and you're comfortable with your ability to add text and graphics to a card, just create the last card titled "What is hardware?" and skip to the next section, "Planting a Hypertext Tree."

What is processing? Intro Details Find Exit

If the computer is used for calculating taxes, the input might be numbers representing wages, salaries, tips, deductions, exemptions, write-offs, and tax tables, and the output might be the number representing the taxes owed.

If it's used for controlling a missile, the input might be radio and radar signals locating the missile and the target, and the output might be electrical signals to control the flight path of the missile.

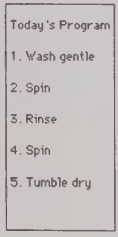
In either case, the computer is following a set of instructions, called a PROGRAM, to transform the input data (information that the computer can read) into the necessary output.



What is software? Intro Details Find Exit

Just as a washing machine goes through a series of steps (wash, extract, rinse, spin) to transform its input (dirty clothes, clean water, detergent) into output (dirty water, clean clothes) a computer follows a set of instructions to transform input into output. These instructions are called SOFTWARE.

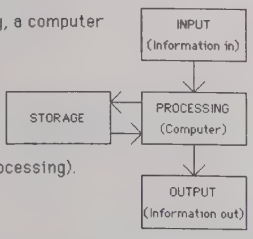
Unlike the washing machine, a computer can operate on many different kinds of input, and be programmed with many different kinds of software. This flexibility is the key to the computer's usefulness. The computer is truly a general-purpose machine.

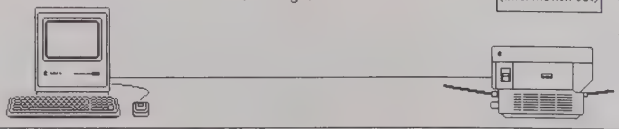


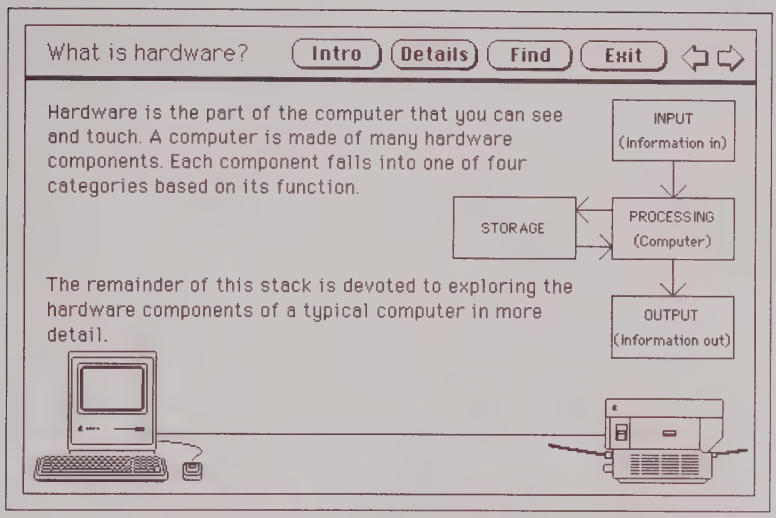
What can it do? Intro Details Find Exit

In spite of its great power and versatility, a computer ultimately can do only four things:

1. Take information in (input).
2. Send information out (output).
3. Do calculations and transformation (processing).
4. Move information around (storage).






What is hardware?
Intro
Details
Find
Exit
Hardware is the part of the computer that you can see and touch. A computer is made of many hardware components. Each component falls into one of four categories based on its function.
INPUT
(Information in)
STORAGE
PROCESSING
(Computer)
OUTPUT
(Information out)
The remainder of this stack is devoted to exploring the hardware components of a typical computer in more detail.

That's enough background for most people. If the whole project were linear like this, we'd have nothing more than an electronic page turner. Now it's time to start branching.

Planting a Hypertext Tree

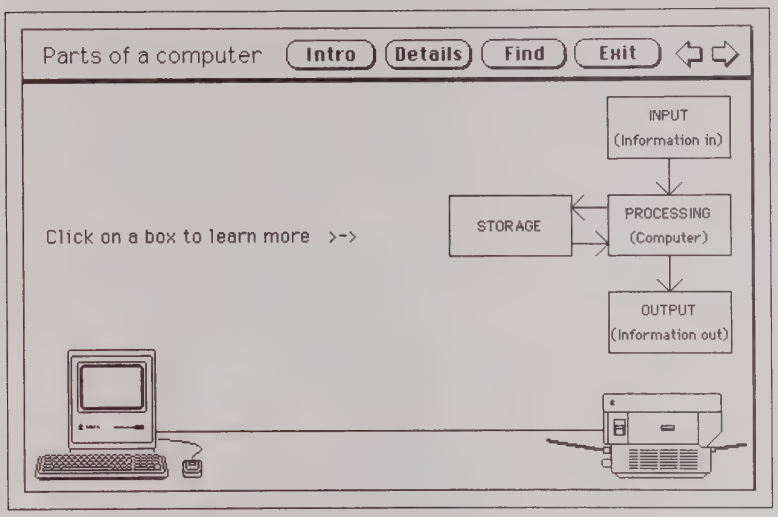
The main body of our hypertext paper is to be tree structured, with buttons that allow the user to follow branches in several directions from a root menu. At the root level, we will offer the user four choices, corresponding to the four component types we've introduced. Why not build buttons into the flowchart we've already created? We can make a copy of the current card and use that as the foundation of our menu card.

Creating a Graphic Button Menu

- ☐ **Type Command-Tab (Browse).**
- ☐ **Select Copy Card from the Edit menu.**
- ☐ **Select Paste Card from the Edit menu.**

You're looking at a new card. It looks like the old one, but not for long.

- ☐ **Using the Browse tool, select and delete the text in both fields.**
- ☐ **In the title field, type "Parts of a computer".**
- ☐ **In the center of the large field, next to the flowchart box labeled STORAGE, type "Click on a box to learn more >->".**



This card is the main title card of the body or the tree of our hypertext paper. So before we go any further, let's link the Details button to this card.

- **Select the Button tool, double-click on the Details button, click on LinkTo..., and click on This Card.**

We want to put transparent buttons on top of the four flowchart boxes. But wait—suppose we put those buttons in the background layer, so they're actually *beneath* the boxes. If we do that, those buttons will work from any card in the stack, provided they aren't covered by other buttons. That means that the user can be instantly transported to any of the four branches at the click of a button. This kind of flexibility fits nicely with our design goals.

There is a small problem with this plan: The buttons will exist even on those cards that don't contain the flowchart. That means that an unsuspecting reader might accidentally click one during the introduction and be transported out of the introduction prematurely. Since we have only a few Intro cards, we can go back later and cover these buttons on the card layer with a large, invisible, do-nothing button. This button will intercept mouse clicks in the area, preventing them from reaching the background buttons. Think of it as an umbrella button. Problem solved.

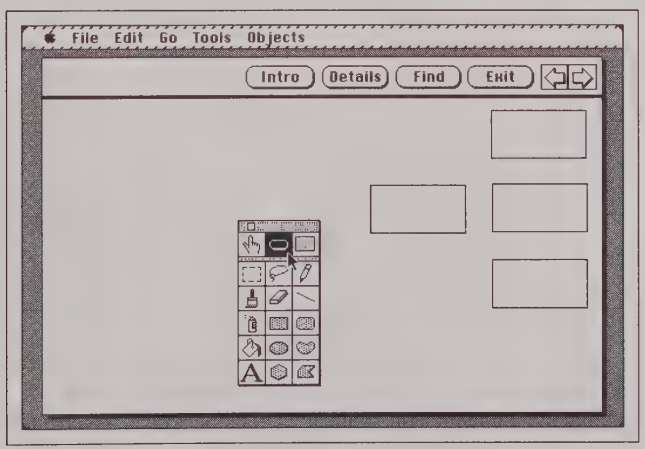
Here's another problem: If we go to the background layer to create the buttons, the flowchart is not visible, so we don't know where to put the buttons. The solution? Temporarily copy the flowchart to the background, and erase the copy when we're finished using it as a template for our buttons.

- ☐ **Using the Selection rectangle, copy (Command-C) the flowchart.**
- ☐ **Type Command-B to go to the background and Command-V (Paste).**

You're in the background, but you have a copy of the flowchart to work with.

- ☐ **Select New Button from the Objects menu.**
- ☐ **Drag and stretch the button so it covers the box labeled INPUT in the flowchart.**
- ☐ **Double-click on the button to bring up the Button Info dialog box.**
- ☐ **Click on Auto hilite, click on Transparent, and click OK.**
- ☐ **Hold down the Option key while dragging this new button to clone it; drag the button clone on top of the box labeled PROCESSING.**
- ☐ **Clone two more buttons for the other two boxes.**
- ☐ **Using the Selection rectangle and the Delete (Backspace) key, delete the temporary flowchart.**

With the Button tool selected, your background should look like this:



- ☐ **Type Command-B to return to the card layer.**

We now need to link each button to a section or branch of cards that deals with the components it represents. We'll eventually need several cards for each branch. Where do we start? Remember that some of our readers may want to simply read every card of the stack in order, from first to last. If we don't plan the order of the cards, sequential readers might find themselves

being jerked from storage to output to CPU to input devices. This kind of unplanned structure is confusing and disorienting, especially for beginners. We should plan a logical order and add cards in such a way that they end up in the order we planned.

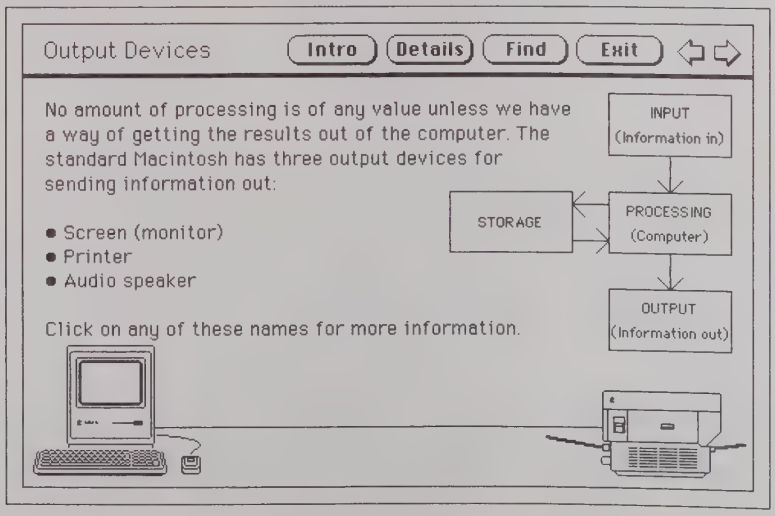
Suppose we want the sequential reader to see the branches in this order: Input, CPU, Storage, Output. What's the easiest way to make sure they end up in that order, given that we'll be creating each of those branches by starting at the main root card? Since the New Card command always inserts the new card immediately after the current card, we can save ourselves a lot of shuffling if we remember this trick:

> When you're creating a tree-structured stack that will also be accessed by some users in sequential order, add branches to the stack in the reverse order that you want them to be read.

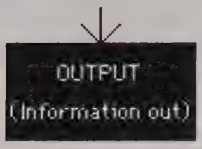
So let's start by creating the Output branch.

Creating a Branch

- ☐ **Select the Button tool and double-click on the Output button.**
- ☐ **Click on LinkTo..., copy the card, paste the card, and click This Card to complete the link.**
- ☐ **Fill the new card with new text, as shown here. Type Option-8 to produce a bullet (•).**



To help readers keep their bearings, we might want to highlight the OUTPUT box in the flowchart.



- ☐ **Carefully select the white area inside the OUTPUT box with the Selection rectangle and choose Invert from the Paint menu.**

We can use this highlighted box throughout the Output branch as a gentle "you are here" reminder.

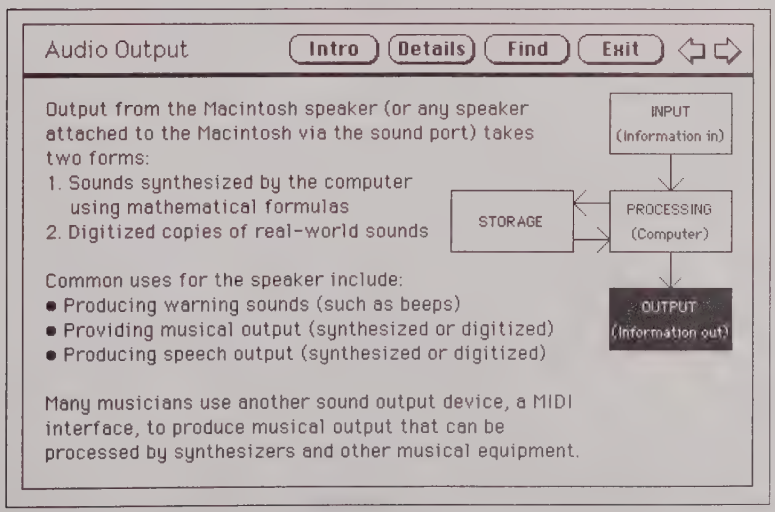
The text on this card instructs the reader to click on the name of an output device for more information. Nothing will happen unless we install buttons on each of those key words.

- Screen (monitor)
- Printer
- Audio speaker

- ☐ **Create a new invisible button, position it over Screen (monitor), and clone it (Option-drag) for each of the other two bulleted key words.**

This is where our Output branch forks. Applying our reverse ordering trick, we should create the last fork of the branch first.

- ☐ **Using the Button tool, double-click on the last of the three buttons (Audio) and link it to a new card.**
- ☐ **Copy the flowchart from the previous card (the one with OUTPUT highlighted) and paste it onto the new card.**
- ☐ **Add text to the new card so it looks something like this:**


Audio Output
Intro
Details
Find
Exit
Output from the Macintosh speaker (or any speaker attached to the Macintosh via the sound port) takes two forms:
1. Sounds synthesized by the computer using mathematical formulas
2. Digitized copies of real-world sounds
INPUT
(Information in)
STORAGE
PROCESSING
(Computer)
OUTPUT
(Information out)
Common uses for the speaker include:
• Producing warning sounds (such as beeps)
• Providing musical output (synthesized or digitized)
• Producing speech output (synthesized or digitized)
Many musicians use another sound output device, a MIDI interface, to produce musical output that can be processed by synthesizers and other musical equipment.

There are plenty of potential hidden buttons on this page. Curious readers might want to know more about synthesized sounds, digitized

sounds, synthesizers, or MIDI. Each of these terms could sit on top of a button linked to a card containing more information on the subject. For small items, we might want to include buttons that reveal pop-up footnotes, rather than jump all the way to another card. We won't do that here, because we've already walked through the process several times. But before you decide to build a stack with hundreds of hidden buttons and links, take a minute to consider a few issues.

The Hazards of Hypertext

Where am I? When links can take a reader to anywhere from anywhere, it's easy for the reader to become disoriented, wondering, "Where am I? How did I get here? How do I get back where I was? How do I get out of here?" Too much freedom can cause anxiety.

What's left? When you finish reading a book, you know you've read every page. Many hypertext documents leave the reader with the uneasy feeling that he or she has missed something important. Some HyperCard documents keep track of the reader's progress automatically, so the reader knows when each session is completed. Until you can build this kind of intelligence into your stacks, it's important to give your stack a structure that makes it easy for the reader to know what he or she has seen so far and what's left to see.

Where are the buttons? When you use invisible buttons behind key words in the text, the casual reader who doesn't know about HyperCard's Command-Option trick might not know where to find them. It's a good idea to highlight those buttons (with shading, all caps, boxes, and so on) to make them obvious.

Are the buttons sticky? If a document has buttons hidden under text, and the text is edited, the buttons don't move with the text. Whenever you edit a hypertext document, make sure that you haven't left buttons under inappropriate words. (The lack of **sticky buttons** in HyperCard is especially a problem when you create **scrolling fields**—fields with scroll bars, allowing large quantities of text to be scrolled through small visible fields. For obvious reasons, buttons can't be associated with words in scrolling fields. (Future versions of HyperCard are likely to contain sticky buttons.)

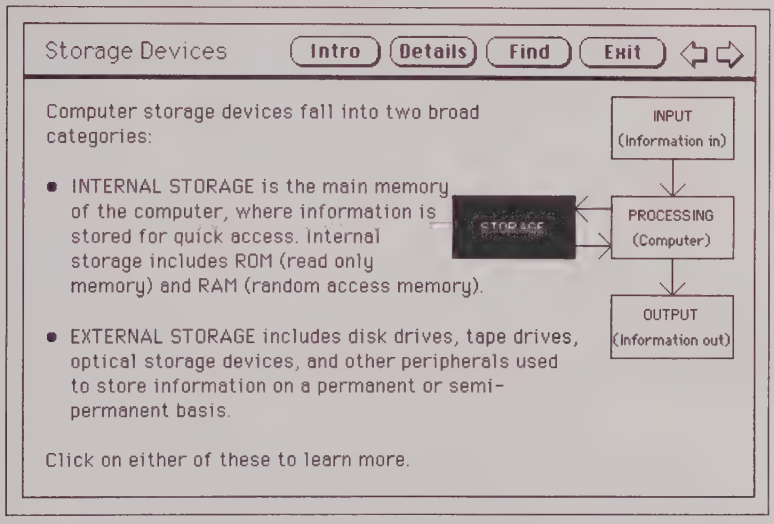
Adding Another Branch

We aren't going to take the time to finish this tree, but we should do one more card in a different branch so we can get a feel for how the finished tree will work.

- ☐ **Return to the Parts of a Computer card by clicking on Details with the Browse tool.**
- ☐ **Using the Button tool, double-click on the Storage button and link it to a New Card.**

This new card is inserted after the root card and before the Output Devices card—right where we want it to be.

- ☐ **Invert the STORAGE box on the flowchart.**
- ☐ **Add text to the fields like this:**


Storage Devices
Intro
Details
Find
Exit
Computer storage devices fall into two broad categories:
• INTERNAL STORAGE is the main memory of the computer, where information is stored for quick access. Internal storage includes ROM (read only memory) and RAM (random access memory).
• EXTERNAL STORAGE includes disk drives, tape drives, optical storage devices, and other peripherals used to store information on a permanent or semi-permanent basis.
Click on either of these to learn more.
INPUT
(Information in)
PROCESSING
(Computer)
STORAGE
OUTPUT
(Information out)

- ☐ **Try the Intro, Details, Storage, and Output buttons from several different locations in the stack.**

As you can see, we don't have a true tree structure, because you can get almost anywhere from almost anywhere else—technically, it's more like a network structure. But the casual user can use the stack like a tree. Readers uncomfortable with the hypertext concept can continue flipping through the stack with the right-arrow button.

Unfinished Business

Keeping Track of Loose Ends

We have much to do before this stack is finished. In truth, a stack like this is never finished, because you can always think of ways to improve it.

It's important to maintain an Unfinished Business list as you create large projects. The human brain can create loose ends much more efficiently than it can keep track of them.

Here's a To Do list that covers the most important missing features. We need to:

- Complete the remainder of the Output branch (screens, printers), remembering to start with the last fork of the branch at any junction. When we reach the leaves on each fork (leaves are cards that don't fork any farther, such as the cards on the ImageWriter printer, color CRT monitor, and so on), we should include pictures of the components being described.
- Complete the other three branches (Storage, Processor, Input).
- Lock the background text fields. We should do this only after we've finished every card with this background, or we won't be able to type in those text fields as we create cards.
- Create do-nothing buttons to mask the background buttons on the Intro cards (and anyplace else they might not be appropriate).
- Embellish the graphics. Some cards, such as the root of the tree, could benefit from a little creative painting.
- Add buttons to the pictures at the bottom. We might want to put invisible buttons behind the screen, printer, keyboard, mouse, and disk drive of the Mac system that sits at the bottom of the root card and link each of those buttons to the card that describes the corresponding component.
- Delete any temporary cards we've created (such as the graphics card we tacked onto the end of the stack).

Finally, the most important step of all:

Test every button and feature of your stack carefully. Test the stack yourself; then ask members of your target audience to test it while you watch.

Critiquing Our Stack

By now, you may be thinking, "This stack has problems; I could design a better one myself." Even when you're designing your own stacks, you're

likely to have those kinds of thoughts while you're turning your design into reality. HyperCard invites experimentation, so stacks tend to evolve as they're being created. Halfway through a project you might come up with a better way to implement a particular feature. Whether you choose to modify your original design or stick with it, it's a good idea to take time to critique your stack to see if it really achieves your original goals.

If we look at this stack critically after testing it ourselves and with novice users, we might start a list of suggested improvements like this:

1. The Exit button is poorly named. Users click it and and expect to leave the stack, but they just go back to the title card.
2. On a related matter, we should reconsider adding some kind of a conclusion. In its current form, the stack is structured like a newspaper article, digging deeper and deeper into details without ever really finishing. Maybe we should keep the Exit button, but link it to a true conclusion, and add another button called Intro.
3. The user interface is clutttered. We have many different kinds of buttons and menus in this stack; this could cause confusion for readers.

This kind of criticism isn't always easy; when you've invested considerable time and energy in a project, it's hard to admit that it has structural flaws. But easy or not, critical analysis is important. Many serious stack designers take the process even beyond critiquing and fine tuning. Here's what those experts advise:

> When you've completed your rough draft of your stack, take an inventory of the ways that it might be improved. Then start over, building it a second time, applying what you learned the first time to your second stack. Repeat as needed.

This is sound advice. It will become especially relevant when you start creating complex stacks with custom HyperTalk scripts. In the next session you'll learn how to do just that.

Summary

You've seen how to create nonlinear hypertext documents using HyperCard. Specifically, you combined several different structures—a sequence, a tree, and a complex network—into a document that can be read in several different ways. You learned how to create and customize text fields, and you learned how to borrow buttons to work with those fields. More importantly, you learned how to apply basic design principles to your work, so the stacks you create can accomplish the goals you've set for them.

Next session you'll begin a journey underneath HyperCard's slick user interface into the fascinating world of HyperTalk. You'll see how this simple but powerful language can be used to add visual effects, animation, and sound effects to your presentations. You'll be amazed at the doors that a few well-chosen HyperTalk words can open for you!

Key Words

branch
Button Ideas stack
Field Info... command
Field tool
Find button
Font/DA Mover
hypermedia/hypertext
Invert command
linear structure
lock text
New Field command
pop-up field
root
scrolling fields
standard fonts
stepwise refinement
sticky buttons
tree structure

Self-Testing Exercises

1. What does this tool do?
2. What happens when a button in the card layer covers a button in the background layer?
3. What happens when a field in the card layer covers a field in the background layer?
4. What happens when a button in the card layer covers a field in the background layer?
5. What happens when a field in the card layer covers a button in the background layer?

6. What kind of structure (linear, tree, or network) is most appropriate for a stack that

   a. provides users with a list of legislators, with options for seeing the voting record of each one?

   b. guides users through a walking tour of your town, highlighting main scenic attractions along the way?

Projects

1. Complete as many of the items on the To Do list at the end of this session as you can.

2. Modify a copy of the Address stack by adding fields for birthday, special interests, and other personal data that might be useful to you. Test the Find command on your modified stack.

3. Create a stack to catalog your collection of tapes, records, or CDs. Include fields for performer, title, music category, favorite tracks, and comments. Borrow buttons to Find and Sort. (The Sort button in the Button Ideas stack sorts according to either the first or second words in the first field you create, asking you to choose whether to sort by first name or last name. It makes the most sense, then, for that to be the performer field.) After entering data for several recordings, test both of these buttons.

4. Modify the Pyramid stack from Session 3 so that it displays informative pop-up fields when buttons are clicked.

5. Modify the Map stack (Project 6, Session 3) so that it displays fields describing particular areas when those areas are clicked.

6. Write an interactive story that allows users to click on buttons to choose different paths through the story.

7. Create a glossary of terms for a specific field of study (computers, hypermedia, biology, anthropology, or whatever), with each term defined and/or illustrated on a single card. Create cross-links that connect related terms, so that readers can jump from definition to definition instantly. For example, if the card for *disk drive* includes the term *peripheral*, that term should have a button that links to a *peripheral* card.

8. Make a genealogy stack with at least three generations. Develop a plan before you actually start to create the new stack. Include in the background five text fields: name, birth date, birth place, interesting information, and date of death. Think about the field attributes carefully in terms of their functions. (Should any of these fields scroll?) Design a good-

looking background; allow space for several card buttons. In the card layer create the following buttons when appropriate: Mother, Father, Spouse(s), and Child(ren). (Why should these buttons be in the card layer?) Link each button to the appropriate card. If the button has nowhere to go, display a message to that effect using a pop-up field. Finally, when you have completed the stack, critique it as if someone else had created it and you are the user.

SESSION 6

Objectives

By the end of this session you should be able to

- Create a slide-show presentation, complete with special visual transition effects between slides
- Create presentations with different kinds of on-screen animation
- Add simple sound and musical effects to HyperCard stacks
- Understand the relationship between HyperCard objects, messages, and scripts
- Locate and understand common HyperTalk scripts in a HyperCard stack
- Write and edit simple HyperTalk scripts

6

BEHIND THE BUTTONS:

EXPLORING HYPERTALK

The Problem

Your planet is in trouble. The climate of the Earth is changing dramatically as a result of short-sighted human activity. You've been asked to give a presentation at a local school assembly, and you'd like to share your knowledge and concerns about this vital issue. You want to back your words up with visual aids that will capture the attention and the imagination of your audience. The school has a device for projecting Macintosh screens so the audience can see them. Perhaps HyperCard can help.

You know how to create cards, illustrate them, add text, and link them with buttons; you're a HyperCard author. But you've seen stacks with features that you don't know how to create: smooth visual transitions between cards, sound effects, even simple animation. You've borrowed buttons that can do some of these things and installed them in your stacks. (That's how you created the pop-up fields, visual wipe effects of the arrow buttons, and logical reference buttons in the last session.) But borrowing buttons, useful as it might be, doesn't tell you what's *really* going on.

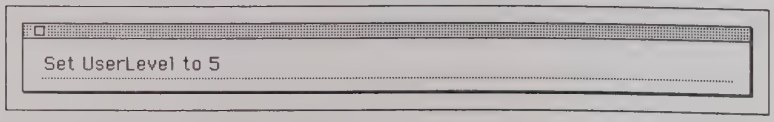
Like an acquaintance from another culture, HyperCard won't reveal its most interesting secrets until you learn a little bit of its language. The native language of HyperCard is **HyperTalk**, a programming language that's about as close to English as any you'll find. In this session you'll learn just enough HyperTalk to give your stacks a professional look and feel. This session is *not* a complete course in conversational HyperTalk; it's more like a workshop on HyperTalk phrases for visitors. After this session you'll be ready to tackle more advanced HyperTalk concepts if the need arises, because you'll have a solid understanding of the basics.

HyperTalk Revealed

Sending HyperTalk Messages

You may not be aware of it, but you've been using HyperTalk already. The **message box** that appears when you issue a Find command is designed for sending messages to HyperCard (and sometimes receiving messages from HyperCard). Those messages must be written in HyperTalk for HyperCard to understand them. So let's start the session by sending a message in HyperTalk.

- ☐ **Start your computer and open HyperCard by double-clicking on the Home icon.**
- ☐ **Type Command-M (Message) to bring up the message box.**
- ☐ **Type this, pressing Return at the end:**



You're speaking HyperTalk now. When you pressed Return, you sent a message to HyperCard to reset the user level to level 5, scripting. If you

don't believe it, go to the last card in the Home stack and see for yourself.

Let's try another HyperTalk message.

☐ **Type:**

As soon as you press Return, HyperCard checks its watch and answers through the message box:

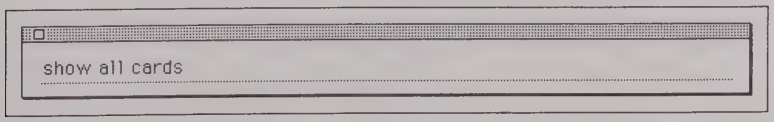
the time is a HyperTalk **function**; when HyperCard sees it, it replaces it with the current time according to the System clock.

The message box can be used as a calculator, too.

☐ **Type:**

HyperCard's answer:

☐ **Here's another interesting command to type:**



The cards in the Home stack fly by after you hit Return, just as if you had clicked on a flying cards button like the one in the Address stack.

- **Finally, type:**

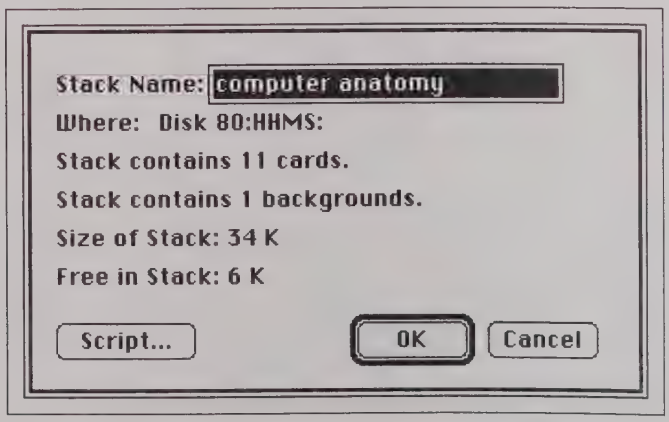
When you press Return, you'll probably see the now familiar dialog box, asking "Where is computer anatomy?" Locate the stack you created last session, use the Drive button, the Eject button, and the HFS maneuvering tools. This has the same effect as using the Open Stack... command.

Examining the Script of a Stack

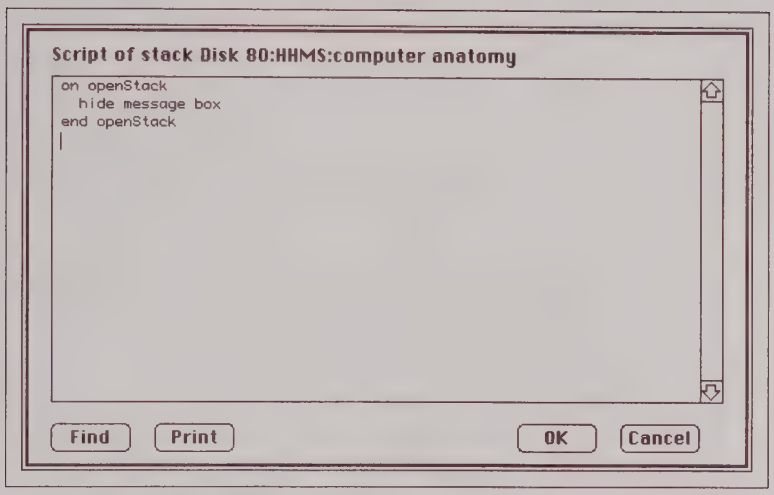
You may be wondering why the message box disappeared when we entered the Computer Anatomy stack. Things like this don't happen accidentally in HyperCard; they happen because someone—or something—tells HyperCard to make them happen. In the first two examples, *we* told HyperCard what to do via the message box. But we didn't tell HyperCard to make the message box go away, so the order must have come from somewhere else.

In this case, the Computer Anatomy stack is the most likely suspect, because the box disappeared when we entered the stack. How can we tell for sure? By checking the stack's **script**.

- **Select Stack Info... from the Objects menu.**



□ **Click on the Script... button.**



There it is, in plain HyperTalk. An English translation of the first line, `on` **`openStack`**, might be "When the stack is opened, follow the instructions between the beginning and the end of this list." The second line, `hide message box`, is the one and only instruction in the list; it tells HyperCard to hide the message box.

> **Show** and **Hide** commands are used to make the message box, the menu bar, the Tools palette, and other objects visible and invisible. As you saw earlier, Show is also used to display a sequence of cards in rapid succession.

If you've ever programmed in a traditional computer language such as BASIC or Pascal, you're probably thinking, "This looks a little like the programming language I've seen before, but there's something strange going on here. Where's the rest of the program?" If you've never seen a computer program before, you're probably just thinking that the whole thing is strange. Either way, a word of explanation is in order—read the Objects and Messages box on the next page.

To help clarify this, let's take a second look at the Computer Anatomy script. The `on` *`openStack`* handler tells HyperCard what to do when the message *`openStack`* is received by the stack. *`openStack`* is a standard message sent by the System to each stack when it is opened. The stack receives the message and says, in effect, "I know what to do with this; my

script tells me to hide the message box when I receive an `openStack` message." HyperCard responds by hiding the message box, and the interaction is over.

Objects and Messages

A program is a set of instructions that tells the computer what to do in the same way that a recipe tells a chef what to do. A program written in a **procedural language** such as BASIC or Pascal is structured like a recipe; the computer works through it, one step at a time, in the order specified by the program.

HyperTalk, in contrast, is an **object-oriented language**. A HyperCard stack is made up of **objects**—cards, fields, buttons, backgrounds, and the stack itself. Each object has a program—a **script**—that gives HyperCard instructions if the object receives a **message**. A script is made up of one or more **message handlers**. Each message handler tells HyperCard what to do if a particular message is received. A message handler takes the form

```
on message
     command(s)
end message
```

Translation: If the message *message* arrives at this object, obey the *command(s)* in this list.

How did the openStack message handler get in the Computer Anatomy stack script? You'll recall that we copied the background for that stack from Art Ideas. You didn't know it at the time, but you were also copying the script for the Art Ideas stack.

But, you might object, this stack does far more than hide the message box. This script doesn't explain enough. The rest of the story is scattered among the buttons, fields, cards, and backgrounds of the Computer Anatomy stack. Let's examine a few of those one more time. This time, though, we'll peel away the HyperCard veneer to reveal the HyperTalk code inside.

Looking Under the Find Button

- ☐ **Click on the Cancel button to hide the stack script dialog box.**
- ☐ **Select the Button tool.**
- ☐ **Select the Find button.**
- ☐ **Select Button Info... from the Objects menu.**
- ☐ **Click on the Script...button in the dialog box.**

Script of bkgnd button id 19 = "Find"

```
on mouseUp
  doMenu "Find..."
end mouseUp
```

This button reveals two more HyperTalk vocabulary words: mouseUp and doMenu.

mouseUp is the most common message for HyperCard buttons. `on mouseUp` is HyperTalk for "If the user lets go of the mouse button while the pointer is on this button, do everything listed between this statement and `end mouseUp`." (While it might seem strange to specify that the action take place when the user lets go of the button, rather than when he or she presses the button, it makes sense. A mouse click consists of two actions—pressing and letting go—and it's not complete until both actions are done.)

doMenu tells HyperCard to execute a menu command. It can be used to automate just about any menu action.

We've seen a stack script and a button script, both of which respond to messages from outside. Stacks, fields, and backgrounds can receive and respond to messages, too. How do all these objects keep their messages straight?

The HyperCard Object Hierarchy

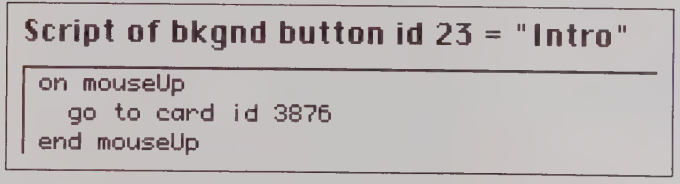
As you point and click your way through a HyperCard stack, you're generating a flurry of messages that are passed upward through the HyperCard bureaucracy. For example, when you click on a button, the System sends a `mouseUp` message to that button. If the button's script has a message handler telling it what to do `on mouseUp`, the button accepts the message and performs the action. If there's no `on mouseUp` message handler in the button's script, the button passes the message on to the next layer—the card, unless there's another button or field on a layer behind this button. If the card has no `on mouseUp` message handler, the message is passed on to the background layer, and then to the stack.

Examining the Script of a Linked Button

Let's try another one.

- ☐ **Click on the Cancel button to hide the Find button's script.**
- ☐ **Select the Intro button.**
- ☐ **Hold down the Shift key while selecting Button Info....**

This is a shortcut to the Script dialog box.



This script is clear, provided you know that the first card of the intro has been assigned ID number 3876. (The number in the script probably won't be the same on your screen, but whatever it is, it's the ID number of the first Intro card. You can check that by going to that card and selecting **Card Info...** from the Objects menu.)

What does this number mean? Nothing; it's just a random ID number used for this kind of thing. HyperCard numbers every object you create; it's up to you to name those objects. By itself, the ID number 3876 tells you nothing about where that card is or what it does; that's why it's smart to give names to the things you create. If a card has a name, you can write scripts that say `go to card intro` instead of letting HyperCard say `go to card id 3876`.

Checking the Script of a Relative Link

Let's check a button with a different kind of script.

- ☐ **Click on the Cancel button to hide the Intro button's script.**
- ☐ **Click on the right-arrow button on the screen and hold down the Shift key while selecting Button Info... from the Objects menu.**

You'll bypass the Button Info dialog box that way, and go straight to the Script.

Script of bkgnd button id 11

```
on mouseUp
  visual effect wipe left
  go to next card
end mouseUp
```

When the mouse is clicked and released on this button, it sends a message to HyperCard to use a **visual effect** called `wipe left` as a transition while going to the next card in the stack.

We'll come back to visual effects soon; for now, notice that there's no mention of a card ID, just a reference to `next card`. This button uses a logical reference, so that it will always go to the next card, no matter what that might be. Compare it with the absolute link in the Intro button. If you click either button on the first card, the result will be the same: You'll go to the beginning of the Intro, which happens to be the second card in the stack. How would these buttons behave differently if we were to insert a card after the title card? (If you don't know, try it!)

The script of the left-arrow button holds no surprises; it's the mirror image of the right-arrow script:

Script of bkgnd button id 10

```
on mouseUp
  visual effect wipe right
  go to previous card
end mouseUp
```

☐ **Examine the script of the Home button.**

Script of card button id 1 = "Home"

```
on mouseUp
  visual effect iris close
  go Home
end mouseUp
```

Another visual effect, and an absolute reference without a number. HyperTalk knows Home by name.

Dissecting a Pop-up Field Button

Let's examine one more script.

- ☐ **Examine the script of the ? button.**

Script of card button id 2 = "help"

```
on mouseUp
  set visible of card field "info" to ¬
  not the visible of card field "info"
end mouseUp
```

This script isn't as obvious as the others we've seen. But remember what the button does, and it's fairly easy to guess what this script is telling HyperCard to do: If the card field "info" is not visible, make it visible; if it's visible, make it not visible. A more accurate translation of the actual script might be, "Make the visibility **property** (called *visible*) of card field info the opposite of what that property is now." (The strange character after `to` means that the command is continued on the next line.) If this is confusing, never mind. HyperCard knows what to do when it reads this, so you don't have to.

Building a Dynamic Presentation

Let's build a stack to make a presentation, and use some simple scripts to give our stack some visual pizzazz.

Outlining the Problem

We know our audience won't learn much if we barrage them with facts and figures, so we have to carefully plan a few main points that we want them to remember. Here's one way to slice up the problem:

I. The greenhouse effect is a problem.

II. We're causing it.

III. It has serious consequences.

IV. We can do something about it.

We need to back each of these points up with some subpoints. We might flesh out our outline like this:

The great geophysical experiment: What are we doing to our planet?

I. What's the problem?

   A. We're changing the Earth's climate.

   B. The rate of change is greater than that of the glacial periods.

   C. We're changing the environment faster than we can understand or predict the consequences.

II. How are we causing the problem?

   A. Burning fossil fuels produces carbon dioxide.

   B. Deforestation (for fuel, grazing land) eliminates carbon dioxide filters.

   C. Human activities produce other greenhouse gases (CFCs, methane, and so on).

III. What are the likely consequences?

   A. Heat waves, drought, and forest fires.

   B. Agricultural losses, famine.

   C. Rising sea level, flooding.

   D. Extinction of many trees, other species.

IV. What can you do?

   A. Stop using Styrofoam and CFC aerosols.

   B. Reduce energy consumption.

   C. Recycle paper, metal, glass, plastic.

   D. Eat less meat.

   E. Plant more trees.

   F. Work for public policies that encourage energy efficiency and discourage pollution.

This outline can be easily transformed into a **bullet chart**, one of the most frequently used forms of **presentation graphics**. Popular software packages such as Persuasion and More II are designed to instantly create bullet charts from outlines like this one. While HyperCard demands more effort, it also allows more flexibility, provided you're willing to speak a little HyperTalk.

Creating an Empty Introduction

- [ ] **Create a new stack called Earth (remember to turn off the Copy current background check box).**

We're going to add an animated introduction later, and that introduction will have a different background than that of the rest of the stack. So let's save a place for the introduction with a single card. (This will save us considerable effort later, since there's no automatic way to insert a new card *before* the current card.)

Creating a Background and a Title Card

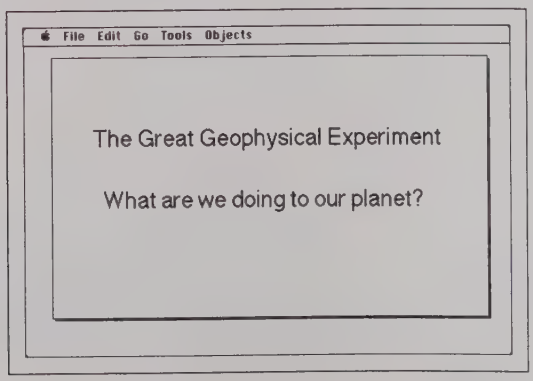
- [ ] **Select New Background from the Objects menu.**

You're now looking at the new background of the second card. We want to put a field in the background, because that's the easiest way to have text in exactly the same location on every card.

- [ ] **Select New Field from the Objects menu.**
- [ ] **Drag and stretch the field so it fills most of the screen.**
- [ ] **Double-click on the field to reveal the Field Info dialog box.**
- [ ] **Click on shadow, Wide Margins, and Font....**
- [ ] **Select a large display font.**

We're using Helvetica 24 point, but Geneva, New York, or Times will work just fine.

- [ ] **Type Command-B to leave the background.**
- [ ] **Select the Browse tool and type a title page, like this:**

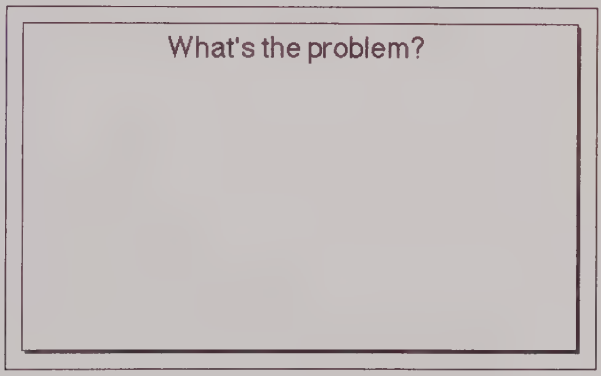


(Since the default justification for the text in this field is left justification, you'll have to center the title by adding spaces in front of it.)

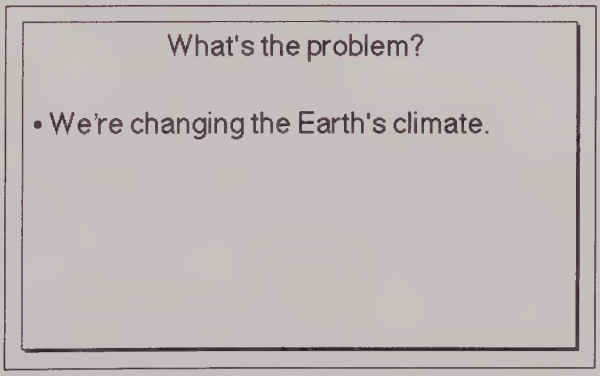
Creating a Dynamic Bullet Chart

We're going to create four bullet charts—one for each main point in the outline. We'd like to have the bulleted points appear on the chart one at a time, as we talk about them. So we'll start with a card containing only the first main heading.

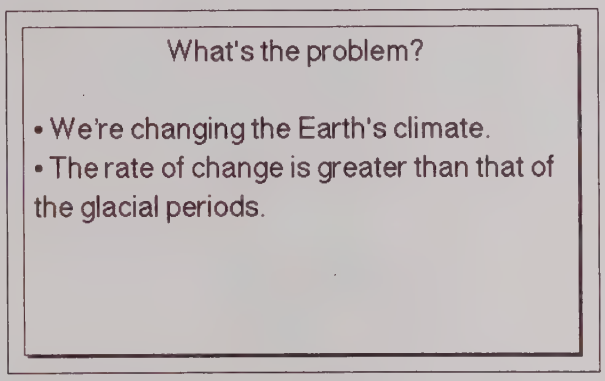
- ☐ **Create a new card (Command-N) and type a title for the top of the card.**

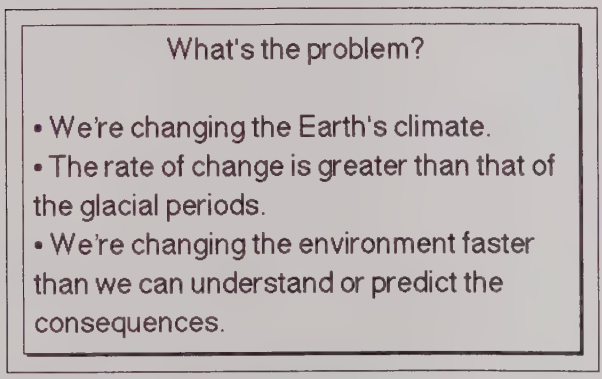


- ☐ **Copy this card and paste it.**
- ☐ **Add another line of text. To type a bullet (•), type Option-8.**



☐ **Repeat the process (Copy Card, Paste Card, type text) for each of these cards:**




What's the problem?
• We're changing the Earth's climate.
• The rate of change is greater than that of the glacial periods.
• We're changing the environment faster than we can understand or predict the consequences.

Building a Background Button with a Visual Effect

Rather than link these cards one at a time, let's create one big background button that will allow us to advance through all of our "slides," complete with a professional-looking visual effect.

☐ **Type Command-B to go to the background.**

☐ **Create a new button. Drag and stretch it to fill the entire screen.**

☐ **Make it transparent and hide the name.**

☐ **Click on Script....**

As you can see, the script doesn't tell HyperCard what to do when the mouse is clicked. It's up to us to type in instructions.

- **Insert the I-beam pointer between** `on mouseUp` **and** `end mouseUp`**, click to position the insertion point there, and type the middle two lines of this script:**

```
on mouseUp
   visual effect wipe right slowly
   go to next card
end mouseUp
```

The Syntax of Scripts

You can type scripts in any combination of upper and lowercase letters; it's all the same in HyperTalk. You don't need to worry about indentation either, because HyperTalk takes care of that automatically. But HyperTalk, like most programming languages, is picky about spelling, spaces between words, order of words, ends of lines, and other matters of **syntax**.

In general, each line of a script represents a single command or statement. If a command is too long to be visible on a single line, you can insert a **soft return** (typed Option-Return, appears in the script as ¬) to tell HyperCard that the command is continued on the next line.

HyperTalk allows enough flexibility that it's possible to write ambiguous statements. Since ambiguity breeds unpredictability, it's important to avoid it whenever possible. Quotation marks (" ") are often used to group words and symbols together to eliminate ambiguity.

Some HyperTalk statements and scripts are clear and unambiguous to HyperCard, but not to people. These statements should be followed by **comments**—clarifying notes for human script readers. A double dash (--) in a script is a **comment marker**; everything that follows this symbol on the line is ignored by HyperCard; think of it as a stick-on note.

If you take liberties with syntax or mistype something, you're likely to see a dialog box with an error message when you try the button. If that happens, go back and compare the script, line by line, with the original, editing the script as needed. Most of the traditional Macintosh editing techniques work, including the Clipboard, provided you use the keyboard shortcuts, rather than the menu, to cut (Command-X), copy (Command-C), and paste (Command-V).

- **Click on OK.**

☐ **Type Command-B (Background), Command-1 (Go First), and Command-3 (Go Next).**

You're now looking at the first card of the presentation (not counting the unfinished introduction).

☐ **Click anywhere on the screen to advance through the slides to the end.**

Notice the smooth transition between cards. *Wipe right* is one of many visual effects you can add to any HyperCard transition.

HyperCard Visual Effects

Well-chosen visual effects can enhance the look and feel of your stack as the user moves from card to card.

The basic syntax of the Visual Effect command in HyperTalk is:

```
visual [effect] effectname
```

The word *effect* is optional. (That's what the brackets mean.) To use the command, substitute an effect name for the italicized placeholder *effectname*.

Examples:
```
visual effect iris open
visual dissolve
```

Visual effect names	Suggested uses:
`dissolve`	for gradual transitions
`iris open`	to go to a card providing more detail
`iris close`	to return after an iris open
`zoom open (or zoom out)`	to go to a card providing more detail
`zoom close (or zoom in)`	to return after zoom open
`wipe left`	to go to next card
`wipe right`	to go to previous card
`wipe up`	to go to next card
`wipe down`	to go to previous card
`scroll left`	to go to next card
`scroll right`	to go to previous card
`scroll up`	to go to next card
`scroll down`	to go to previous card
`barn door open`	to introduce a topic
`barn door close`	to end a topic
`checkerboard`	between cards with high contrast
`venetian blinds`	between cards with high contrast

You can add an optional speed specification after the effect name, provided you arrange the command in this order:

```
visual [effect] effectname [speed]
```

Example: `visual effect barn door close very slowly`

Speed choices:
```
slowly
very slowly
quickly
very quickly
```

You can also specify the screen image HyperCard will use during the transition:

```
visual [effect] effectname [speed] [to image]
```

Example:

```
visual effect checkerboard slowly to black
```

Image choices:

`black`	
`white`	
`gray`	
`card`	(the image of the destination card)
`inverse`	(the reversal of the card image)

Finally, you can include several visual effects in a row; they'll all happen in a sequence at the next Go command. Experiment!

Rearranging Layers in the Background

Now it's time to add another card. You should currently be looking at the last card of the bullet chart sequence you just created.

- ☐ **Type Command-N (New Card) and move the pointer over the background field.**

Something is wrong; the pointer doesn't turn into an I-beam. Why? Because the full-screen button we created is standing between our pointer and the field, blocking our attempts at typing. But we can move the field in front of the button with another useful command.

- ☐ **Select the Field tool, click on the background field, and select Bring Closer from the Objects menu.**

Creating More Bullet Charts

We have three more complete slides worth of information to type.

- ☐ **Create the following cards.**

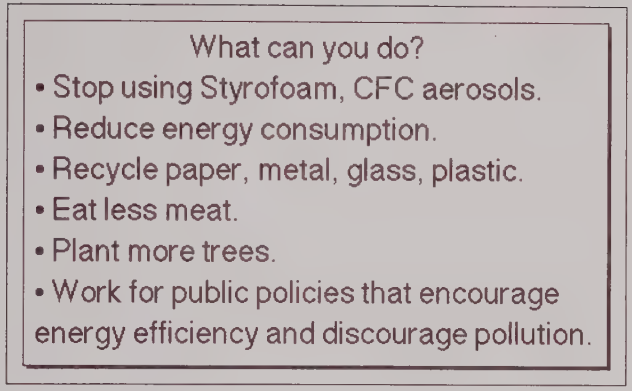
If you're in a hurry, just create one of these cards. If you want to build a complete presentation, break each of these into several cards so the complete bullet chart builds one point at a time, the way we did the first one.

How are we causing the problem?

- Burning fossil fuels produces CO2.
- Deforestation (for fuel, grazing land) eliminates CO2 filters.
- Human activities produce other greenhouse gases (such as CFCs and methane).

What are the likely consequences?

- Heat waves, drought, and forest fires.
- Agricultural losses, famine.
- Rising sea level, flooding.
- Extinction of many trees and other species.


What can you do?
• Stop using Styrofoam, CFC aerosols.
• Reduce energy consumption.
• Recycle paper, metal, glass, plastic.
• Eat less meat.
• Plant more trees.
• Work for public policies that encourage energy efficiency and discourage pollution.

Adding a Card Layer Illusion

- ☐ **Go to the title card (the second card, The Great Geophysical Experiment), select the Button tool, and copy the background button. (You don't need to go to the background to copy this button, since there are no buttons in front of it.)**
- ☐ **Paste it onto the card layer of this title card.**

The background button is still there, but on this card it's completely covered by the new card button. When the mouse is clicked on this card, the card button will intercept a *mouseUp* command before the background button can receive it. We can now change the script of the card button so that it has a different visual effect.

- ☐ **Hold down the Shift key and double-click on the button.**

Shortcuts to Scripts

To see a button's script	Shift-double-click with button tool selected, or press Command-Option and click on the button, or press Shift-Command-Option and click on the button
To see a field's script	Press Shift-Command-Option and click on field
To see a card's script	Command-Option-C
To see a background's script	Command-Option-B
To see a stack's script	Command-Option-S
To see any script after an error message	Click on the Script... button in the dialog box

- **Edit the script just as you'd edit any Macintosh text; change the visual effect from** `wipe right` **to** `scroll down`.
- **Try the button.**

The scroll is effective for creating the illusion that we're moving the current card out of the way so we can see the next one, just as the wipe creates the illusion that we're writing text on the card. If you decide to complete the bullet chart presentation, copy this button onto each *completed* bullet chart card (showing every point), excluding the final "What can you do?" card.

Once all the text is typed and checked, there's one more step:

- **Using the field tool, select the background field; select Send Farther from the Objects menu to return it to its original layer.**

Hiding the field behind the button will provide some security against accidental deletion of the text. We don't need to lock the text in the field, because there's no way to type in a buried field.

Moving Pictures

Creating a Starry Background

Now it's time to add some full-blown **animation** at the beginning to capture the attention of the audience. The simplest kind of animation in HyperCard is familiar to every child: flipping through a stack of cards with slightly different pictures on them—card-flipping animation. Let's draw a series of

pictures zooming in on Earth from a distance. When the Earth gets close enough to be easily recognized, we'll make it dramatically go up in smoke.

Start with a background of stars.

- ☐ **Type Command-1 (Go First) and Command-B (Background).**
- ☐ **Use the Paint Bucket to paint the background black.**
- ☐ **Select All of the background and use the Lighten command in the Paint menu to produce a random star pattern like the one on p. 202.**
- ☐ **Make a tiny white mark in the background halfway down the screen near the left side just big enough to distinguish it from the stars).**

Adding a Growing Planet

Geography, Maps

- ☐ **Go to the Geography, Maps card of the Art Ideas stack and copy the six Earths all at once using the Lasso and Command-C (Copy).**

- ☐ **Go back to your stack.**

You should be looking at the card layer of the first card.

- ☐ **Type Command-N (New Card) twice.**

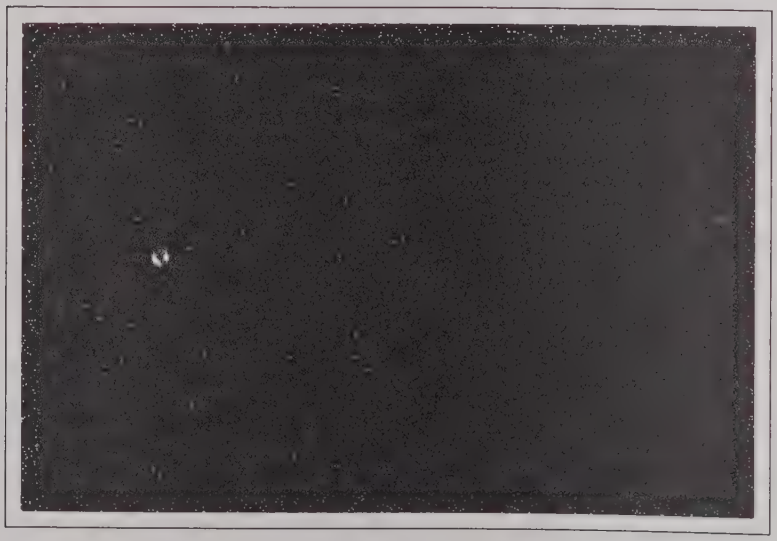
You're looking at the third card in the stack.

- ☐ **Type Command-V (Paste Picture).**

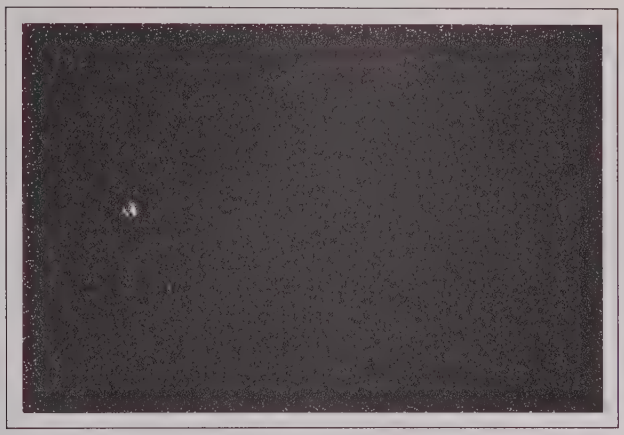
All of those Earths you lassoed are now on the card.

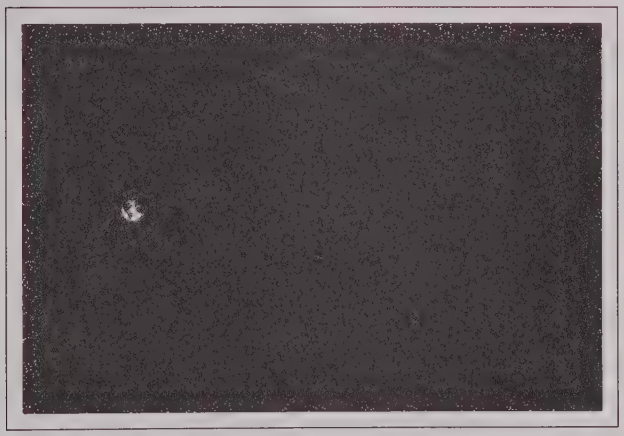
- ☐ **Lasso the smallest one, move it over the small mark you made, and delete the others (using the Lasso and Backspace/Delete).**

- Create a new card (Command-N), paste (Command-V) the Earths again, lasso the next smallest one, move it over the mark, and lasso/delete the others.



- Repeat with all the other sizes of the Earth, choosing a bigger size each time.

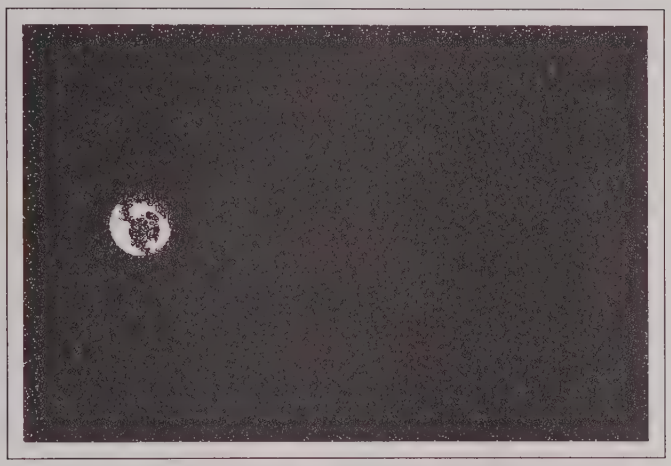




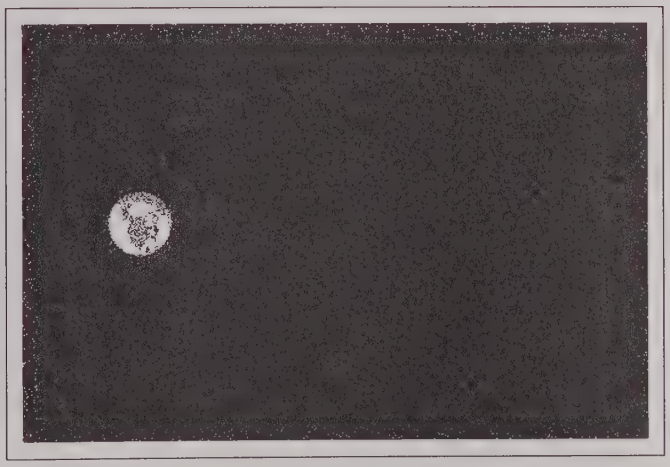
- ☐ Use the Paint Bucket to fill in the continents of the largest Earth, so it's consistent with the other ones.

- ☐ When you complete this card, select Copy Card and then Paste Card.
- ☐ On the new card, select the Earth with the Lasso and use the Lighten command from the Paint menu several times.

- ☐ Copy the card with the lightened Earth, paste it, and lighten the Earth several more times.



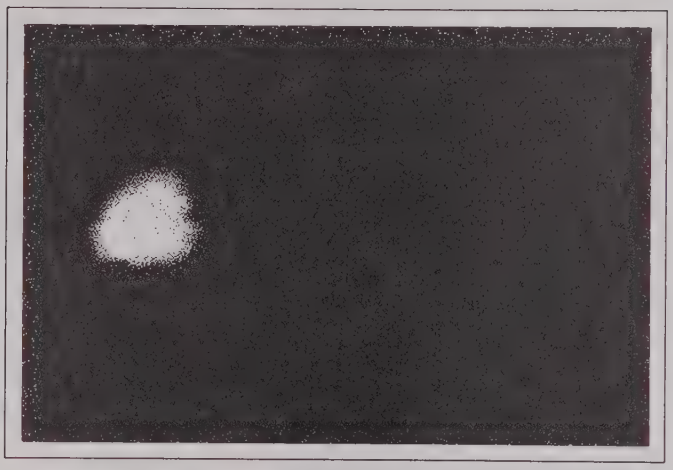
- [ ] **Repeat the last step one more time (Copy Card, Paste Card, Lighten).**



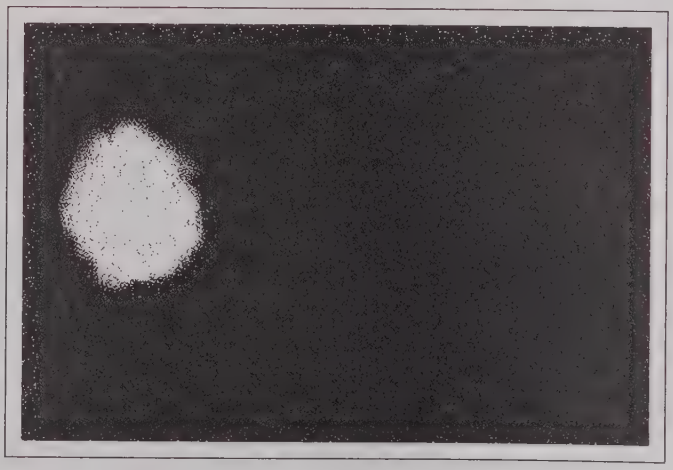
Spraying Away Life as We Know It

Now you get to destroy the Earth with (what else?) an aerosol can.

- [ ] **Copy the current card, paste it, select the Spray Can and white paint, and spray in small circles around the Earth until it looks like a cloud. (While you're at it, experiment with the Paint Brush with different brush shapes and sparse patterns.)**

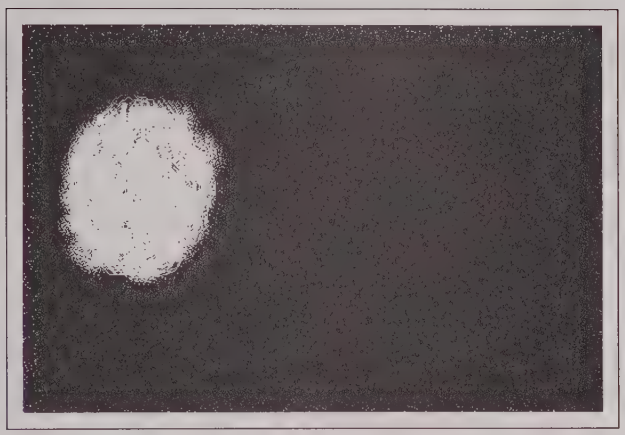


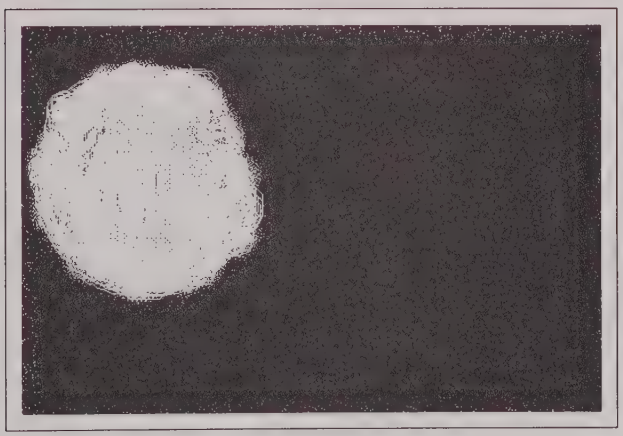
- **Copy the current card, paste it, and enlarge the cloud.**

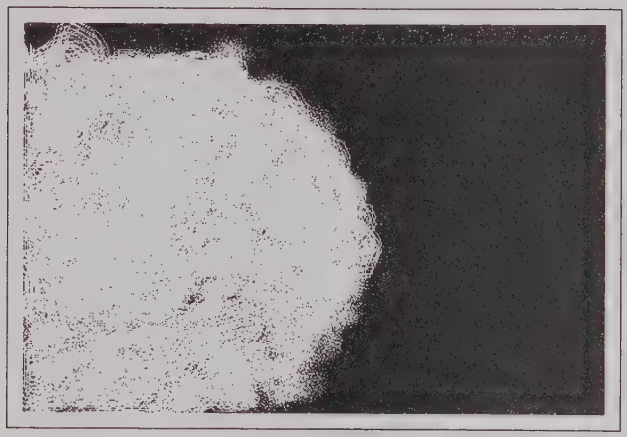


- **Repeat the process several times, until the cloud fills the screen. Remember to paint behind the menu bar on the larger clouds, because we're going to turn it off later so our presentation fills the whole screen. (Type Command-Space to hide the menu bar and to bring it back later.)**

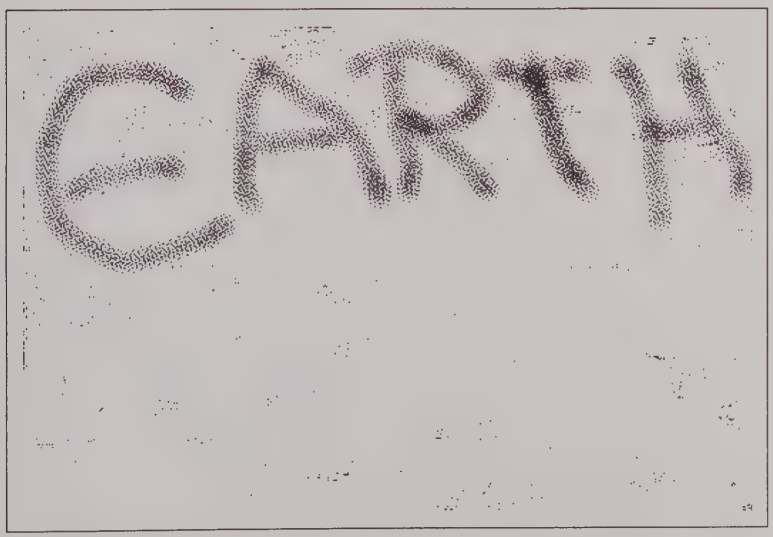
Do as many as you want. The more frames, the more convincing the animation.





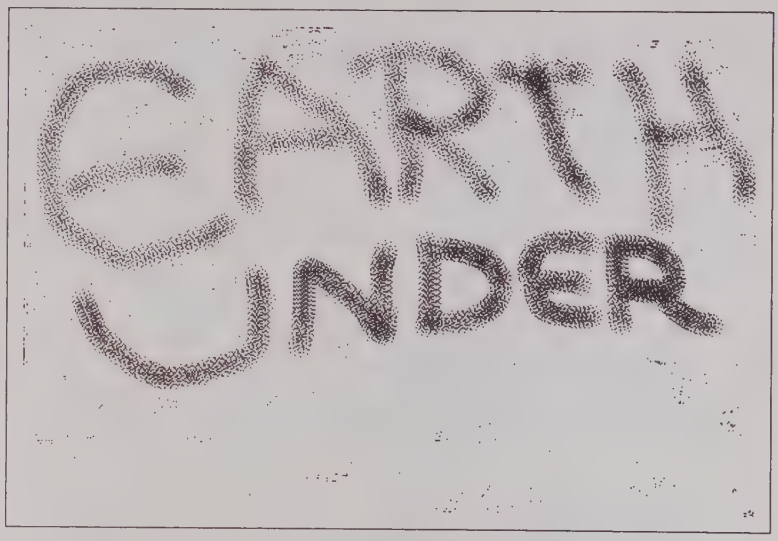


☐ **Copy the full-screen card and paste it; use the Spray Can to write:**

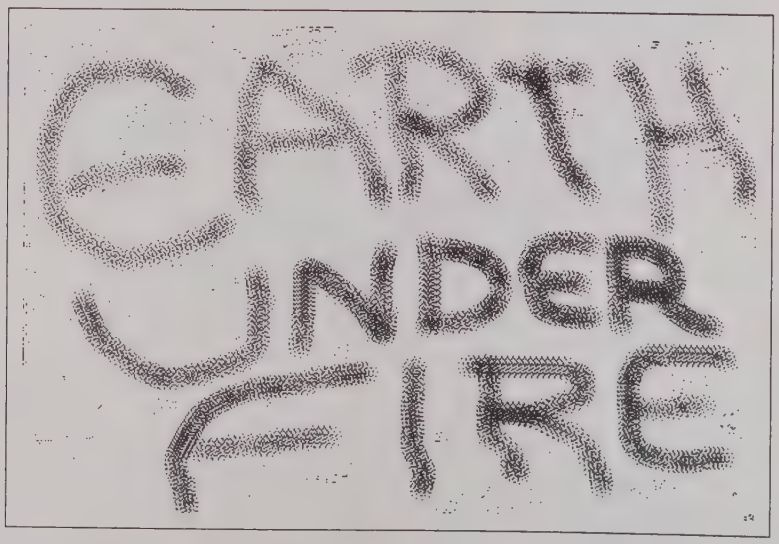


Remember that you can use the Revert command to erase the entire word if you aren't happy with it.

☐ **Copy and paste the card and add another word:**



☐ One more time:



Scripting an Animated Sequence

☐ Try using the arrow keys to flip through these cards.

The animation is a little jerky. We could link these cards, one at a time, and add scripts to dissolve from one card to another to smooth the transitions. But there's an easier, more effective way. We can put one button on the first card to start the show and add a script to that button that causes *all* of these cards to be displayed in order, with dissolve effects before each one. To top it off, we can add a statement to hide the menu bar so it won't distract our audience. Here's how:

- **Go to the first card in the stack (Command-1).**
- **Create a new button, stretch and drag it to fill the screen, and double-click on it to bring up the Button Info dialog box.**
- **Click on transparent, Show Name (to turn it off), and Script....**
- **Insert the I-beam between** `on mouseUp` **and** `end mouseUp` **and type the remaining lines of this script:**

```
on mouseUp
   hide menuBar
   repeat 22 times
      visual dissolve slowly
      go to next card
   end repeat
end mouseUp
```

Depending on how many cards you have in your sequence, you might need to adjust the number in the Repeat statement. Here's a rough translation of the script: "When the mouse is clicked, hide the menu bar and repeat 'dissolve to next card' 22 times."

> The **Repeat** command repeats everything between `repeat` and `end repeat` the specified number of times.

- **Select the Browse tool and click on the button to try it.**

If the button works properly, the resulting sequence should be dramatic: The Earth gradually comes closer, then turns to a smoke cloud that overtakes the whole screen, revealing the title of the presentation, one word at a time. If it doesn't end on the correct card, you'll need to change the number in the Repeat line. When it's working properly, you'll find yourself staring at the Earth Under Fire card when the sequence ends.

Linking the Introduction to the Presentation

We need a button on this card to dissolve into the framed title card.

- ☐ **Create a full-screen, invisible button on the Earth Under Fire card with the following script:**

```
on mouseUp
   visual dissolve slowly
   go to next card
end mouseUp
```

Creating Cards for a Finale

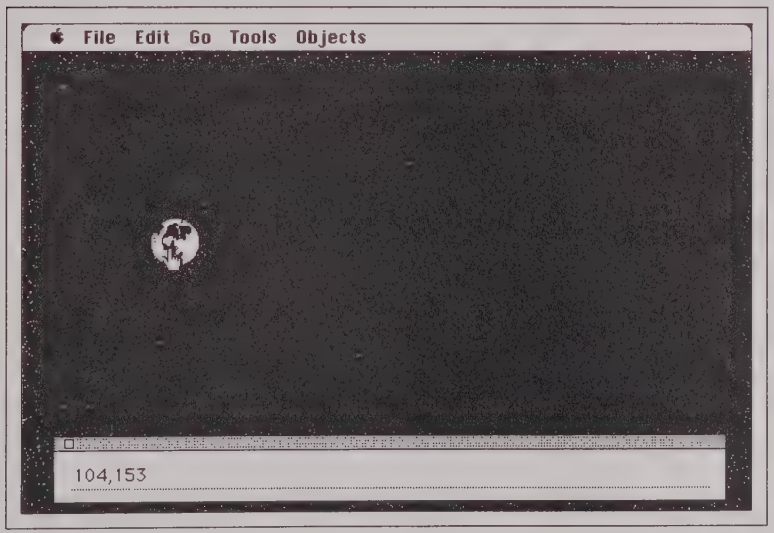
Let's use another kind of animation—**paint tool animation**—to punctuate our conclusion.

- ☐ **Go to one of the cards in the introduction with a large image of Earth and select Copy Card.**
- ☐ **Type Command-4 (Go Last) and Command-V (Paste Card) *twice*.**
- ☐ **On the last card (the one you're looking at after the last Paste command), use paint text and a large display font to type the sentence shown here. To get the white-on-black effect, use the Selection rectangle to select the white area around the text and select Invert from the Paint menu. (If the text still doesn't show, type Command-S to select all card-layer painted objects; then select Opaque from the Paint menu.)**

Finding the Mouse

- ☐ **Position the pointer so that it's fingering Earth.**
- ☐ **Type Command-M (Message) and type:**

You just used **the mouseLoc function** to ask HyperCard where the mouse pointer is on the screen. When you press Return, HyperCard responds in the message box with a pair of numbers:



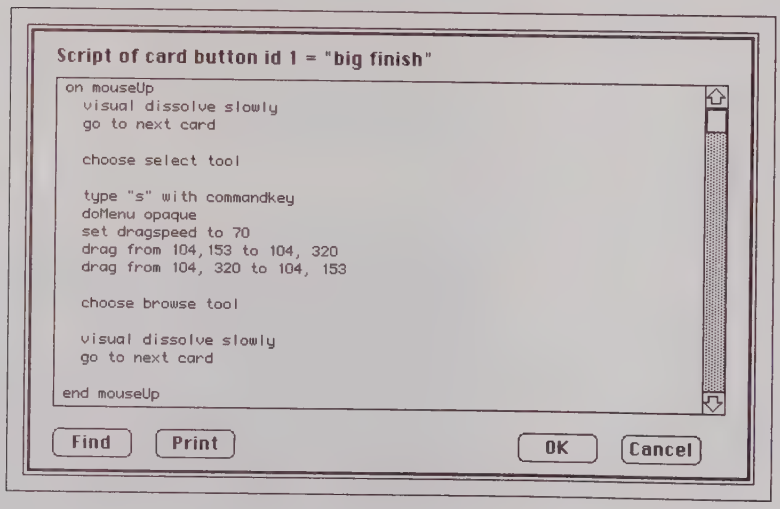
The screen is made up of a 512 x 342 grid of pixels; each pixel has a unique address defined by two numbers: distance in pixels from the left, and distance in pixels from the top. The two numbers in the message box are the coordinates of the current mouse position. You'll need those numbers soon.

Scripting an Automatic Lasso Button

- ☐ **Type Command-2 twice to go back to last bullet chart card.**

☐ **Create a full-screen invisible button and type the following script, substituting numbers as follows: (1) Replace every occurrence of 104 in the script with the first number in your message box; (2) replace every occurrence of 153 in the script with the second number in the message box.**

(By making those substitutions, you're telling HyperCard exactly where on your screen the Earth image is.)



This script is telling HyperCard to perform this sequence of actions:

1. Dissolve slowly to the next card (the first of the two new cards).
2. Choose the selection tool (the marquee) from the Tools menu.
3. Type Command-S to select everything painted in the card layer (in this case, only the Earth is selected).
4. Select Opaque from the Paint menu to make the Earth opaque, if it isn't already.
5. Establish a speed for the Earth to be dragged through space.
6. Drag the selected object from its current position on the screen to the bottom of the screen and back.
7. Turn the pointer back into the Browse tool.

8. Dissolve slowly into the next card—the one with the white-on-black concluding message.

In essence, you told HyperCard to lasso the Earth and drag it down and back. In fact, just about any HyperCard menu command can be automated using HyperTalk scripts.

Note: this script will not work unless the user level is set to 3 (painting) or higher. If you want to play it safe (usually a good idea when writing scripts), you could insert `set userLevel to 3` —or 4 or 5—immediately after `on mouseUp`.)

You've tried three kinds of animation commonly used in HyperCard: visual effects, card flipping, and paint tool animation. Visual effects are the easiest, because HyperTalk is programmed to do the work for you in response to a single command. Card-flipping animation isn't difficult to create from a series of slightly different cards, but it can gobble up disk space in a hurry, since each card is stored as a separate picture. Paint animation requires a more analytic approach to scripting, but it's useful for certain kinds of effects.

HyperTalk programmers have devised other types of animation that are well suited for particular applications. We won't cover those types of animation here, but Appendix A, HyperCard Sources and Resources, contains more information on animation resources, including useful tutorials and two programs for creating more professional animation effects that can be controlled by HyperCard.

Singing Stacks

This is an impressive finish for our presentation, but something is missing: sound. When you know a little scripting, it's easy to add scripts that can make noise or play notes. Most musical scripts are attached to buttons, but we're going to set up a script to play a borrowed bit of Bach automatically when the final animation sequence begins.

Making a Musical Card

☐ **Go to the second to last card (the one with the Earth alone) and select Card Info... with the Shift key held down.**

This card has no script, so it's up to us to type one.

☐ **Type the following script:**

Script of card id 17693

```
on openCard
  repeat 5 times
    play "harpsichord" tempo 150 "g4e f g eb g d g c"
  end repeat
end openCard
```

openCard is a message, like `openStack` and `mouseUp`. An `openCard` message is sent by the System when the card is opened. When that happens, this message handler tells HyperCard to repeat the play command 5 times.

The key to this script is the **Play command**. If you can read standard musical notation, this command is fairly easy to interpret once you understand how it works. If you're not a musician, the command is, at best, cryptic. Either way, this brief description of the command may make its function clearer.

The Play Command

The play command takes the following form:

play "*voice*" [tempo *tempovalue*] "*notes*"

Here's a brief explanation of each element of the command:

voice tells HyperCard what kind of sound to play. `Harpsichord` and `boing` are built into HyperCard; if you want other sounds, you'll have to add **resources** (data) for those sounds using one or more of the tools described in Appendix A, HyperCard Sources and Resources.

tempovalue tells HyperCard how fast to play the sequence of notes. 100 is normal; larger numbers represent faster tempos.

notes that make up the melody to be played may be represented by their letter names (a through g), separated by spaces and followed by four optional characters: (1) accidentals (*b* for flat or # for sharp); (2) octave number, with 4 representing the octave for middle c; (3) the duration of the note (*w* for whole, *h* for half, *q* for quarter, *e* for eighth, *s* for 16th, *t* for 32nd, and *x* for 64th); and (4) . for dotted note or 3 for triplet.

Everything except accidentals stays in effect until it is changed, so you don't need to repeatedly specify note length or octave. In the example,

```
play "harpsichord" tempo 150 "g4e f g eb g d g c"
```

all of the notes are eighth notes (`e`) in the fourth octave.

HyperCard doesn't stop working while a sound is playing, so it's possible to combine audio and video effects to create simple multimedia presentations.

(Sounds may be distorted if you're not running from a hard disk system; many other systems simply can't access information fast enough for distortion-free sound.)

Those are the basics, but if you're serious about sound, there's much more to learn. Consult Appendix A, HyperCard Sources and Resources, for more sound information, including how to put your own real-world sounds into stacks.

Smart Links

Anticipating an Interactive Presentation

The Earth stack is a powerful audiovisual aid for a presentation, with its dramatic introduction and conclusion sandwiching the point-by-point bullet chart display. But it's a linear structure, designed to be viewed from beginning to end without interruption. Suppose someone in the audience asks for a graphic explanation of the causes of the greenhouse effect or demands that you support your premises with solid quantitative evidence? If you can instantly respond by displaying a relevant drawing or a chart, you're more likely to win support for your cause.

As long as your stack is strictly linear, like conventional presentation aids, you can't provide immediate visual responses to questions because you don't know when those questions will occur. Where in the stack should you put a drawing or chart providing answers? What if no one asks the questions your visual aids are anticipating?

One solution is to include an answer button at the bottom of each bullet chart; this button can take you *temporarily* to a card (or card sequence) containing visual answers to the most likely questions—perhaps an animated diagram showing the way carbon is exchanged between the atmosphere and the earth, a chart showing the rise in average global temperature, or a sequence that includes both. At least one of these answer cards will need to include a button to take you back to the bullet chart sequence so you can continue your presentation after the question is answered.

Obviously, the "take me back" button can't be hard-linked to any single bullet chart card; it needs to be flexible enough to return your exact point of departure, *wherever that might be*, so that you can continue the presentation from where you left off. The return arrow button is often used for this kind of job. But the return arrow button is not telepathic; it won't work properly unless the point of return has been marked in advance. In this example, the answer button on the bullet chart card can mark that card before transporting you to the answer card(s).

The answer button and the return button will work as a team, like this: You're making your presentation, clicking on the top of the screen each

time you present a new point. Somebody in the audience says he wants more details on how the greenhouse effect works. You click on the hidden answer button at the bottom of the screen. This button effectively does two things: It makes a note of the current card (in the same way you might write down your page number before closing a book) and it jumps to another card containing a diagram. You use the diagram to answer the question and then click on a return button. The return button checks the note that was left by the answer button (just as you might check your page number before returning to your book) and returns to the marked spot.

That's the theory; here's how to make it happen.

Providing Many Happy Returns

- ☐ **Go to any of the bullet chart cards and type Command-B to go to the background.**
- ☐ **Create a new button and drag/stretch it so that it covers a thin strip across the bottom of the screen.**
- ☐ **Make the button transparent, turn off the Show Name option, and name the button "answer button".**
- ☐ **Type this command between** `on mouseUp` **and** `end mouseUp` **in the button's script:**

```
push card
```

When this button is clicked, HyperCard will push a reminder onto its stack of reminders in memory; this reminder will tell HyperCard where to return when it's told later to pop a card off of the stack. This kind of **stack** is not the same as a HyperCard stack; it's a special kind of **data structure** that allows information to be recovered in the opposite order that it was stored, so that the last item added to the stack is the first one removed. The conventional metaphor for a stack data structure is a spring-loaded stack of cafeteria trays, but a more appropriate metaphor for this example might be an old-fashioned spindle for holding small paper notes. In any case, we aren't going to allow reminders to pile up here, so we can ignore the terminology and the metaphors altogether and think of this simply as a way of allowing HyperCard to remember where to return when we click on a return arrow button.

> When HyperCard encounters a **`push card`** command, it remembers the location of that command. The next time it encounters a **`pop card`** command, it returns to that point.

- **Double-click on the button and click on LinkTo....**
- **Type Command-4 to go to the last card in the stack.**
- **Select New Background from the Objects menu.**
- **In the LinkTo... dialog box, click on the button that says This Card.**

We just added a new card with a plain background to the end of the stack. This card, and any that follow, will contain the charts and other visual tools that can be used to support arguments and answer specific questions from the audience. We won't create the graphics for this card. (If you want to add some yourself, you might want to look for ideas in the September, 1989 issue of *Scientific American.*) We *will* add a button to this card to take us back where we came from.

- **Type Command-4 to go to the last card — the blank card you just created, and type Command-B to go to the background.**
- **Create a new button and drag it to the lower right corner.**

- **Name the button "Return", make it transparent, turn off the Show Name option, and attach a return arrow icon to it.**
- **Insert this line in the button's script between** `on mouseUp` **and** `end mouseUp`**:**

```
pop card
```

There's no need to link this card to another card. When it's clicked, this button tells HyperCard to return to the card that was visible the last time a `push card` command was issued.

- **Return to any of the bullet chart cards and try the combination of push and pop buttons to make sure that they work properly.**

Bringing It Home

We have two more tiny scripts to write.

- **Put an invisible full-screen button on the last card and give it the following script:**

```
on mouseUp
      go Home
end mouseUp
```

- **Type Command-Option-S to show the script for the *stack*, and write a three-line script:**

```
on closeStack
      show menuBar
end closeStack
```

Always put things back the way you found them; if you hid the menu bar when the stack opened, show the menu bar when you close it; if you changed the user level on `openStack`, change it back on `closeStack`; and so on.

Missing Links

HyperTalk Debugging and Housekeeping

HyperTalk is a rich and powerful language with capabilities that go far beyond what we've covered here. You've had enough of a taste of HyperTalk so that you're well equipped to explore the language further. HyperTalk is the kind of computer language that encourages experimentation, and experimentation is a wonderful way to learn. But as you're exploring, remember that HyperTalk, like any powerful tool, can cause problems as well as solutions.

Avoiding HyperTalk Hazards

- Use the Save a Copy... command regularly when you're working on an important stack. The more often you back up your work, the less work you'll lose if something goes wrong.
- Use the **Compact Stack** command after you've been working on a stack for a while. This command gets rid of pockets of **free space** that develop whenever you delete anything; this free space makes the stack slower and larger. (You can check the amount of free space in the Stack Info... dialog box.)
- Plan ahead, and don't be afraid to backtrack or start over if necessary.
- Develop the habit of naming everything with meaningful names, and write scripts that `Go to` *`name`* rather than `Go to` *`number`*. It's much easier to debug a script full of names than it is to debug a script full of numbers.

- Avoid duplicate names; there's nothing in HyperCard to prevent two things from having the same name, so it's up to you.

- Use quote marks to keep things together that belong together.

- Make your scripts as readable as possible, and add comments anywhere that the meaning or purpose of a statement isn't obvious.

- Use temporary scripts to help you debug your stacks. Here's a useful example: When you're creating a stack with many links going in different directions (such as Dungeon or Computer Anatomy), it's easy to lose track of the ordering of the cards. Add a field to each background, check its Field Info dialog box to see what number has been assigned to it, and create a background script like this:

```
on openCard
  put "Card" && number of this card into field 1
end openCard
```

  (Translation: when the card is opened, **put** a character string that's made up of (1) "Card" and (2) the number of the current card. `&&` tells HyperCard to combine those two elements with a space between them (`&` would combine them with no space). In some stacks, such as Computer Anatomy, this field might be useful as a permanent fixture. In other stacks, such as Dungeon, where you don't want the card number displayed, the script should be removed when the stack is finished.

- Put scripts, buttons, and fields where they are most appropriate. If a script, button, or field affects only one card, put it on that card. (Or, if it's a script, put it in a button or field of that card). If it affects many cards with the same background, put it in the background. If it affects many cards with different backgrounds, put it in the stack. Use a consistent method for deciding where to put scripts, so you can easily find them later.

- Don't reinvent the wheel unless you have to. Many commercial, shareware, and public domain programs are filled with buttons, cards, and stacks designed for borrowing. If somebody has created a script that does what you want, and has given the world permission to copy it, feel free to do so. (However, it's important to respect copyright laws and the ethical underpinnings of intellectual property rights; don't borrow from copyrighted material without the author's permission.)

- Use the right tool for the job. HyperTalk is an **interpreted language**, which means that the HyperTalk **interpreter** translates it on the fly, like an interpreter at the United Nations. Programs written in **compiled languages**, in contrast, are translated all at once into machine language by **compilers**, the way this book might be translated from English to Spanish. The tradeoff is between immediacy and speed; since it's an interpreter, HyperTalk produces relatively slow-running programs. If your HyperCard stack needs speed—or some other feature or property that's not native to HyperTalk—you might be able to have the best of both worlds by including an **XCMD (external command)** written in Pascal or C in your stack. Writing XCMDs requires considerable programming skill, but borrowing XCMDs is easy. Many XCMDs are available in commercial and public domain packages.

- Learn from the masters. Study the best stacks that you can find and apply the programming techniques of those stacks to your own work. (A sampler of first-rate stacks is included in Appendix A, HyperCard Sources and Resources.)

The Endless Stack

The scripts you've seen and written in this session should give you a feel for the basic mechanics of HyperTalk. You'll be surprised how much more you can learn by snooping in scripts of other people's stacks, consulting the Help stack to look up commands that are unfamiliar. The Help stack is designed to be a reference tool, not a tutorial, so you may want a more systematic introduction to HyperTalk. Appendix A lists several such sources, as well as software tools specifically designed to facilitate the stack-building process.

There's no denying the power of HyperTalk as a tool for enhancing your scripts. But even if you never learn another word of HyperTalk, you'll be able to apply what you've learned in this book to build impressive and useful stacks. You now have the tools to make things happen. Go for it!

Summary

In this session you went behind HyperCard's friendly front and learned a few words of HyperTalk, the scripting language of HyperCard. You learned how to send simple messages to HyperCard in the message box. You saw how the objects of HyperCard—buttons, fields, cards, backgrounds, and stacks—are controlled by scripts that interpret messages and execute commands. You used HyperTalk to add visual effects, animation, and music to a stack. In the process, you learned enough about HyperTalk scripting to easily learn more by examining other scripts, the HyperCard Help stacks, and other HyperTalk resources.

Key Words

animation
Bring Closer command
bullet chart
Card Info ... command
card-flipping animation
comment (in script)
comment marker (¬)
Compact Stack command
compiled language
compiler
doMenu command
free space
function
Hide command
Hypertalk
interpreted language
interpreter
Lighten command
message
message box
Message command
message handler
mouse up
object
object-oriented language
openCard
openStack
paint tool animation
Play command
pop card command
push card command
Put command
presentation graphics
procedural language
property
Repeat command
resources
script
Send Farther command
Show command
soft return (Option-Return, ¬)
stack data structure
Stack Info...
syntax
the mouseLoc function
the time function
visual effect
XCMD (external command)

Self-Testing Exercises

1. What do these script commands do?

```
visual wipe right
play "boing" c
repeat 10 times.... end repeat
drag from 104,153 to 104,320
go to next card
```

   Where might you use each one—button, field, card, or stack scripts? Describe the reasons for your decisions.

2. What happens when you type Command-M?
3. What is the difference between an object and a message?
4. Why does a button script generally contain the phrase `on mouseUp`?
5. What is syntax and how is it important in HyperTalk?
6. List three visual effects, what they do, and suggested uses for each one.
7. What does `the mouseLoc` mean?
8. Describe three methods of HyperCard animation.

Projects

1. In the background of the bullet chart portion of the Earth stack, create a thermometer shape along the left edge of the screen. If you like, put a clip-art Earth in the bulb at the bottom of the thermometer. On the card layer, draw a line representing the temperature on the thermometer. Have it rise a little on each successive card, until you get to the "What can you do?" cards. Have the temperature drop for each new point on these cards.
2. Create buttons on your Home card that will automatically open each of the stacks you've created.
3. Add visual effects and sound effects to the Pyramid stack, the Dungeon stack, and the Computer Anatomy stack.
4. Modify the Dungeon stack (from Session 4) so that the bed's button causes the bed to move slowly, revealing the secret passageway underneath as it moves. Use card-flipping animation techniques. Copy the first card showing the bed and paste it. Remove the two buttons from this new card. Use the Lasso to select the bed and move it a small amount. Copy the new card and move the bed again. After you've created cards

showing the complete bed-moving sequence, work *backward* through these cards, adding the pit in successively smaller segments to each card. Create a button on the first bed card to flip through the cards. (You will need to know how many cards are in the series when you script the button.) For a final touch, have the Bed button make a boing sound when you click it.

5. Create a card containing musical buttons. Make each play a different note. Using paint tools, create a keyboard and put the buttons on each note. Play a tune.

APPENDIX A

HYPERCARD SOURCES AND RESOURCES

HyperCard in a Hurry was designed to give you a quick but thorough introduction to HyperCard. If you decide to work seriously in software development with HyperCard, you'll want to learn more about HyperCard and other tools that extend your powers as a stack builder. This appendix surveys software and hardware tools for adding graphics, animation, sound, scripts, and other enhancements to your stacks. In addition, it describes an assortment of exemplary stacks available through commercial and public domain channels so you can see how professionals and serious amateurs are putting HyperCard to work. Finally, this appendix lists several books that you can use to expand your knowledge of HyperCard and HyperTalk.

Graphic Tools

A paint program can be a useful accessory for HyperCard authors, HyperCard paint tools notwithstanding. There are two reasons why you might want to use a paint program with HyperCard:

1. HyperCard's Import Paint... command can access only the top left part of a standard Macintosh paint document, so most clip art pictures and digitized images have to be edited and manipulated with a paint program before they can be imported into HyperCard.

2. Many paint programs have tools and options that go far beyond HyperCard's painting capabilities. By creating your images in one of these programs and importing the results into HyperCard, you can often save time and produce more artistic stacks.

Several popular paint programs are available for the Macintosh. Here are three worth knowing about:

- MacPaint (Claris) was originally designed by Bill Atkinson, the creator of HyperCard, so it shares many of the same features, tools, and menus. If you're comfortable drawing in HyperCard, MacPaint will feel like home from the minute you double-click it. Unfortunately, the aging MacPaint is short on features when compared with almost every other paint program on the market. Except for its full-page and multiwindow capabilities, it has little to offer that you don't already have in HyperCard.
- Studio/1 (Electronic Arts) isn't as easy to learn to use as MacPaint; you'll probably have to spend some time with the manual before you can make sense of the many unfamiliar tools and menu options. But if you'll be creating many HyperCard stacks with complex graphics, Studio/1 is worth getting to know. Modeled after Studio/8, the powerful color paint program from the same publisher, Studio/1 provides an array of tools and tricks that make the toolboxes in HyperCard and MacPaint look conspicuously underpowered. Custom brushes, dithered gradients, editable Bezier curves, 3-D perspective, free rotation, and a host of other features allow you to quickly create graphic effects that would take hours in HyperCard. As if that weren't enough, Studio/1 is a marvelous tool for building animated sequences of painted pictures that can be displayed from within Studio/1 or HyperCard. (See the next section, "Animation Aids," for more details.)
- DeskPaint (Zedcor) provides many of the sophisticated painting tools of Studio/1 in the convenience of a desk accessory. Because it's a DA, DeskPaint is always available in the Apple menu. This makes it easy to edit and import clip art, scanned images, and other paint files without leaving HyperCard. If you don't have enough memory to run HyperCard and Studio/1 simultaneously under MultiFinder, DeskPaint is an alternative worth considering. (DeskPaint also includes DeskDraw, an object-oriented drawing DA, and an auto-trace capability that allows you to convert bit-mapped graphics to object-oriented graphics for smoother printouts.)

With a good scanner it's possible to copy art from books, magazines, and other noncomputer sources. Inexpensive scanners can turn photographs and line drawings into digitized images that can be edited and imported into HyperCard. Scanners vary widely in price and features. Here are three of particular interest to HyperCard users:

- Apple Scanner (Apple). This is one of many gray-scale flatbed scanners on the market. (Flatbed scanners read face-down pages like a photocopier does.) It's capable of scanning images at much higher resolution than you can see on the Macintosh screen. Its distinguishing feature from the point of view of a HyperCard user is the included HyperScan software that allows you to control the scanner directly with HyperCard.

- ThunderScan (Thunderware). This low-priced device turns an ImageWriter printer into an input device by temporarily replacing the ribbon with a photoelectric cell that can pick up images from paper fed through the printer. It's dreadfully slow and not always accurate with fine detail, but it's priced at a fraction of the cost of a flatbed scanner and it does the job.

- LightningScan (Thunderware) and ScanMan (LogiTech). Between ThunderScan and the Apple Scanner in price and performance, these two hand-held scanners are similar in appearance, capability, and price. Each scanner can quickly and easily pick up images almost 5 inches wide from any printed page. The main differences between these two units are in the software. Since software revisions are promised soon, you'll make the most informed choice if you try both units before you buy.

If you don't have a scanner and you aren't an artist, you'll probably be able to put clip art to good use. If you can't find what you need in the public domain, you'll probably find it in a commercial collection. High quality bit-mapped art packages are available in collections such as Wet Paint (Dub'l Click Software), Click Art (T-Maker Software), and Mac the Knife (Miles Computing). There are dozens of other collections; *Macworld* magazine does an exhaustive annual review of clip art collections.

Animation Aids

In Session 6 you learned how to animate HyperCard stacks using visual effects, card flipping, and paint tool animation. These animation effects, along with button animation, font animation, cursor animation, and other animation tricks, are illustrated in the public domain (free) stack by Jeremy Ahouse called Animation Help (a *Macworld* SuperStacks award winner), and in an inexpensive stack from Heizer Software called HyperCard Visual Animation Tricks. Both of these stacks are excellent resources for learning more about HyperCard animation. As your proficiency in HyperTalk increases, you'll be able to copy and modify the scripts in this stack and use them to animate your own stacks

Although HyperCard offers plenty of possibilities for animation it places clear limitations on the artist or programmer interested in serious

animation. Fortunately, several software tools are designed for creating works that can easily be displayed from inside a HyperCard stack, such as:

- Studio/1 (Electronic Arts) combines a powerful paint program (see the previous section) with an easy-to-use animation builder. In order to display Studio/1 animated sequences (complete with sound effects) in HyperCard, you need to install a series of external commands (XCMDs), but Studio/1 comes with a stack that automates the installation process. These XCMDs provide HyperCard with thevocabulary to open Studio/1 files and play them without leaving HyperCard (provided the Studio/1 files are in the same folder as the current stack). With a minimal amount of scripting, you can create on-screen buttons that control the speed and direction of the animation the way infrared remote buttons can control a VCR.

- Director (MacroMind) is the most popular high-end animation package for the Macintosh. Director is far more than a simple animation package; it's really a multimedia program that stretches the limits of the Macintosh hardware. A series of XCMDs make it possible to play Director movies from within HyperCard (if the movie file and the MacroMind Player file are in the same folder as the current stack). On a Mac II, these XCMDs allow you to display full-screen moving *color* pictures from within HyperCard!

- ScreenRecorder (Farallon) is a special-purpose tool for recording and playing back Macintosh screen movements. It's an ideal tool for anyone creating Macintosh tutorials. It comes with a HyperCard stack that makes it possible to play back the screen recordings from within HyperCard.

Sound Resources

HyperCard comes with two sounds built in: boing and harpsichord. Many stacks have buttons to play other sounds, ranging from symphonic strings to Sylvester and Tweety Bird. It's possible to copy those buttons and paste them into your stacks, but the copied buttons won't play the sounds in the new stack because they lack resources. Every sound in HyperCard is stored as part of a resource. (Resources are used to store sounds, icons, XCMDs, and other data used by HyperCard and other programs.)

Resources can't be copied like buttons; they have to be moved with special software tools. The most widely available program for moving resources is ResEdit, a tool used by Macintosh software developers for creating and modifying resources. But ResEdit is

dangerous in the hands of the uninitiated, because it gives you the power to destroy a stack or a System with the click of a mouse. A safer and friendlier tool for copying resources is Steve Maller's public domain program called ResCopy. ResCopy allows you to copy sound resources from other stacks into your stack. Once you've done that, you can play those sounds in a variety of ways with the Play command.

A growing number of stacks take care of the resource copying automatically. One of the most popular is HyperComposer, a *Macworld* SuperStacks award winner published by Addison-Wesley. HyperComposer contains dozens of musical instrument sounds that can be imported into your stacks at the click of a button. HyperComposer also takes much of the tedium out of creating HyperCard melodies. With HyperComposer, you can enter a melody using either a musical staff or an on-screen keyboard, edit the resulting HyperCard melody script using standard Macintosh editing techniques, and export the melody to your own stack, including the instrument resource, with the click of a button.

Probably the most important tool for anyone serious about HyperCard sound is Farallon's MacRecorder. MacRecorder is a combination of hardware and software that allows you to capture any sound as a digitized Macintosh sound resource that can be used in HyperCard, Studio/1, Director, ScreenRecorder, or other programs. The mouse-sized hardware device attaches to the modem port of the Macintosh; it can be connected to any audio source, or it can be used with a microphone to capture real-world sounds. HyperSound software that comes with MacRecorder makes child's play of recording sounds and porting them directly into your stacks as ready-to-push buttons. SoundEdit software allows you to add echoes, play the sound backwards, bend it, change the tempo, or modify it in other ways before installing it in your stack. With MacRecorder and a little creativity you can add a powerful audio dimension to any of your stacks.

There's just one catch: Sounds take up *lots* of space. An otherwise tiny stack can expand to fill a floppy disk with the addition of just a few digitized sounds. MacRecorder allows you to compress sounds so they take up less disk space (with some cost in fidelity), but even compressed sounds add significant storage overhead to a stack. So if disk space is at a premium, take some sound advice: Choose and use sounds sparingly.

HyperTalk Helpers

It's not necessary to master every detail of HyperTalk to produce professional-looking scripts, provided you have a thorough understanding of basic stack construction techniques and a willingness to learn from others. Many software tools are available to assist you as you build stacks, so you don't need to do everything the hard way. Some tools are power tools to make the script-writing process easier and to improve the quality of finished stacks.

Others are packages of prefabricated buttons, art, cards, and XCMDs that can easily be incorporated into stacks. Here are some examples:

- ScriptEdit (Somak Software) is a powerful desk-accessory text editor for streamlining the process of writing and editing scripts. An Objects window displays a dynamic list of every object on the current card; a double-click on any object in that list opens a window for editing the related script. You can edit several scripts at once using ScriptEdit's powerful editing features and the Macintosh Clipboard. ScriptEdit is a real time saver for serious scripters.

- CompileIt! (Heizer Software) is a compiler to convert HyperTalk scripts into XCMDs or XFCNs (external functions). CompileIt! can be useful for speeding up overall stack performance, protecting scripts so they can't be bootlegged, and saving time on character manipulations and integer calculations. It also provides HyperCard access to the Macintosh ToolBox ROM routines. CompileIt! is easy to use if you understand scripting techniques, but it requires considerable technical know-how to take full advantage of it.

- Port Authority (Heizer Software) is a useful utility for transferring files between HyperCard and other applications, including many popular word processors, spreadsheets, and database programs.

- 101 Scripts and Buttons for HyperCard (Individual Software) is a diverse collection of HyperCard tidbits, including animated buttons, menu buttons, text-processing utilities, music utilities, and visual effects.

- XCAL (Trendware) adds several XCMDs to HyperCard that make it easier to create spreadsheet-style stacks.

- Dialoger (theResult Software) provides XCMDs to simplify the process of creating true Macintosh-style dialog boxes.

In addition, there are hundreds of public domain and shareware tools, buttons, cards, and art collections available via electronic bulletin boards and user groups. Some of these are of questionable quality, but there are also treasures aplenty for those who are willing to dig.

Prime Examples

Imitation may be the sincerest form of flattery, but it's also a great way to learn. Thousands of stacks have been created since HyperCard was introduced in 1987. Many of these stacks have modest price tags; others are

available free through public domain software channels. By snooping through some of these stacks, studying their design and analyzing their scripts, you can learn tricks and techniques that you aren't likely to find in books. Here's a list of stacks that illustrate a variety of HyperCard applications and techniques.

Bird Anatomy (public domain, available from Berkeley Macintosh User Group). A comprehensive study of bird anatomy, complete with sound effects. This is a good model for educational stacks.

Desktop Publishing Advisor (Broderbund Software). This is two stacks in one: a hypertext (and graphics) tutorial of the fundamentals of DTP design principles and a project management system for tracking and organizing DTP projects.

Events Day by Day (Heizer Software). There's nothing fancy here; just a day-by-day database of famous births, deaths, and events. Still, this stack illustrates the power of HyperCard for providing almost instant answers to questions like "When was Gandhi born?" or "What interesting things happened on this date?"

Focal Point II (TENpoint0). If you're intrigued by the way HyperCard can be used to organize your address book, appointment calendar, and business projects, you'll appreciate Danny Goodman's definitive time- and task-management package. Even if you never use it, this stack can teach you all kinds of invaluable scripting techniques.

The Freedom Trail (public domain, available for a small fee from the Boston Computer Society, 617/625–7080). This *Macworld* SuperStacks award winner was created by volunteers of one of the largest and most active Macintosh user groups in the world. If you're interested in American history, Boston, or state-of-the-art educational hypermedia, study this stack carefully.

HyperDA (Symmetry Corporation). This desk accessory allows you to open and use HyperCard stacks on a standard Macintosh without quitting other applications.

HyperTutor (teligraphics). An interactive HyperTalk tutorial. If you want something more engaging than a book, this may be just the right tool for expanding your knowledge of HyperTalk.

If Monks Had Macs (public domain, available from Berkeley Macintosh User Group). A massive hypertext tour de force that's both inspiring and

mind boggling. A crisscross catalog of ideas with a medieval flavor.

Inigo Gets Out (available in the public domain or as part of Amanda Stories, Volume I, from Voyager Software). This stack established HyperCard as a new medium of children's literature, using the kinds of techniques you used to create the Dungeon game in Session 4.

A Legacy of Graphic Design (Heizer Software). A winner of the *Macworld* SuperStacks Contest and a highly graphic sampler of modern design styles.

The Manhole and Cosmic Osmo (Activision). If Inigo Gets Out established the genre, these two programs define the current state of the art. These massive stacks are fun for kids of all ages. The Manhole has an Alice-in-Wonderland kind of magic that's 100 percent HyperCard, whereas Cosmic Osmo is more of a science-fiction cartoon spoof with lots of VideoWorks animation sequences seamlessly woven in.

Monster Hunt (shareware) and Murder on Main Street (Heizer Software). These two game stacks for older players build on the techniques you learned when building the Dungeon.

Star Myths (Dr. Brian L. Crissey, Linfield College Computing Science, McMinnville, OR 97128). This beautiful stack shows what a class of dedicated college students can do with a good idea, a roomful of Macs, and HyperCard. This stack is a fascinating vehicle for exploring the connections between ancient cultures and the night sky.

State-Smart (HyPerFormance, P.O. Box 1591, Corvallis, OR 97339, 503/758-3429). A comprehensive collection of U.S. maps and facts, with all kinds of cross-links that allow you to browse wherever your curiosity leads you.

The Vision of HyperCard, or Bill Sez by Bill Atkinson (public domain). This is a short audio-visual slide show of HyperCard's creator discussing the underlying philosophy at the 1984 Hacker's Conference.

User Groups, Bulletin Boards, and Information Services

Many of the best HyperCard stacks, clip art collections, and other useful software tools are available for free or with a pay-if-you-like-it understanding. User group libraries, electronic bulletin boards, and information services are good sources for public domain software and shareware.

User groups have different policies on public access to their libraries and bulletin boards. The Berkeley Macintosh User Group (BMUG, 1442A Walnut St. #162, Berkeley, CA 94709, 415/849–9114), one of the largest user groups in the world, maintains a large, well-organized collection of public domain and shareware programs on disks that may be purchased by members and nonmembers for a nominal fee. Their occasional catalogs are excellent tools for sorting through the mountains of free and almost-free software for the Macintosh.

If there's a Macintosh user group in your area, they can probably offer you far more than a software library. User groups can be a valuable source of answers, ideas, and other support that would be difficult to package on a floppy disk. To locate the user group nearest you, ask your local Apple dealer or call Apple's User Group Hotline: 800/538-9696, ext. 900.

There's at least one user group specifically for HyperCard fanatics. It's had an on-again-off-again history since HyperCard's release. For information on its current status, contact Apple HyperCard User Group, 20525 Mariani Ave., Cupertino, CA 95014, 408/ 974-1707.

Heizer Software's Stack Exchange (800/888-7667, or 415/943-7667) isn't really a user group, but it does provide a valuable service by acting as a clearinghouse for inexpensive stacks. The quality of Heizer stacks is, on the average, higher than that of public domain and shareware collections, but the cost is considerably lower than most commercial packages.

Information services are also a good source of stacks, software tools, and HyperCard information. But because most of these services charge several dollars per hour of connect time, they're not usually the cheapest source. Two of the most popular nationwide services with Macintosh forums are CompuServe (800/848-8199 or 614/457-0802) and GEnie (800/638-9696).

Other HyperCard Books

No single book can tell you everything you need to know to take full advantage of HyperCard. Each book listed here has something to offer for anyone serious about HyperCard stack development.

The Complete HyperCard Handbook, by Danny Goodman (New York: Bantam Books, 1987). This book was released the day HyperCard was introduced to the world, and it's still the most popular HyperCard book on the market. A very readable introduction to HyperCard and HyperTalk, complete with lots of sample scripts.

HyperCard Developer's Guide, by Danny Goodman (New York: Bantam Books, 1988). This book takes up where *The Complete HyperCard Handbook* lets off, covering advanced stack design, advanced HyperTalk concepts, resources, and XCMDs.

HyperCard Script Language Guide: The HyperTalk Language (Reading, Mass.: Addison-Wesley, 1988). This is the official Apple guide to HyperTalk. It's much more technical and thorough than *HyperTalk Beginner's Guide*, and it's a valuable reference for serious HyperTalk programmers.

HyperCard Stack Design Guidelines (Reading, Mass.: Addison-Wesley, 1989). This official Apple publication covers the human side of stack development. If you care about the user interface of your creations, read this book carefully.

HyperTalk, by Lon Poole (Richmond, Wash.: Microsoft Press, 1988). This is an inexpensive, pocket-sized reference guide for programmers who have learned the basics.

HyperTalk Beginner's Guide: An Introduction to Scripting (Cupertino, Calif.: Apple Computer, 1989). If you bought a Mac recently, this book probably came with it. If you're interested in learning more about scripting, this is an excellent sequel to *HyperCard in a Hurry*.

XCMD's for HyperCard, by Gary Bond (Portland, Ore.: MIS: Press, 1988). If you know Pascal or C, HyperTalk, and a fair amount about Macintosh programming, this book will show you how to write XCMDs that can be included in your stacks.

APPENDIX B

PREPARING YOUR COMPUTER SYSTEM: A LAB ADMINISTRATOR'S GUIDE

To use HyperCard and this book, you'll need a Macintosh with a minimum of 1 megabyte of memory. If you're using a 1-meg system, make sure the RAM Cache is turned *off* in the Control Panel. Use HyperCard version 1.2 or later (at press time, the current version of HyperCard is 1.2.5). Free upgrades are available from Apple dealers and many other sources.

Hard Disk

HyperCard works best when installed on hard disks. To best take advantage of your resources, copy HyperCard, Home, Help Stacks, Idea Stacks, and More Stacks from the Apple HyperCard disks onto your hard disk, keeping files organized in their original folders. The single exception: Make sure that HyperCard and Home are in the same folder, or that neither is in a folder. Make sure that the System file on the hard disk has the Calculator, the Chooser, the Control Panel, and several sizes of Geneva font installed.

File Server

A file server is a good alternative for labs that don't have hard disks on all machines. If you're administering a lab file server, copy HyperCard, Home, Help Stacks, Idea Stacks, and More Stacks from the Apple HyperCard disks onto the server, making sure that all of these files are flagged as read-only. Make sure that HyperCard and Home are in the same folder, or that neither is in a folder.

Diskette

If you're using a machine with two standard double-sided (800K) disk drives, you may be doomed to an eternity of disk swapping. Your only hope is to use one drive for a personal system disk and one drive for a HyperCard/Home disk. As long as there's room, save your files on one or both of these two disks, rather than having a third disk for data.

If you're using a Mac SE with three drives (two internal and one

external), use one for the System, one for HyperCard and Home, and one for data.

If your machine has one or more high density Superdrives and you have HD disks, prepare an HD HyperCard disk that contains the System, HyperCard, and Home. Use the other drive for HyperCard data files.

The latest version of the Apple HyperCard User's Guide has an extensive section of tips for using HyperCard without a hard disk.

APPENDIX C

MACINTOSH SURVIVAL SKILLS

Working with Disks

To start the computer: turn it on, if necessary insert a system disk
To initialize (format) a new disk: insert the disk, click Two-Sided, click Erase, type a name, click OK
To erase an old disk: insert and select the disk, select Erase Disk from the Special menu
To eject a disk: select disk, select Eject from the File menu, or (if necessary) press Command-Shift-1 or Command-Shift-2
To remove a disk from desktop and memory: drag the disk icon into the Trash icon (also ejects the disk)
To copy a disk: drag the source disk icon onto the destination disk icon
To lock (write protect) a disk: slide the plastic tab to open the hole (close to unlock)
To open a disk window or folder window: double-click the icon *or* click the icon and select Open from the File menu
To close a disk window: click the close box *or* click the window and select Close from the File menu

Working with Folders

To create a new folder (from the Finder): select New Folder from the File menu; type **folder name**
To move a file or folder into a folder: drag the icon into the folder icon

To move a file or folder out of a folder: drag the icon out of the folder window into the destination window

To open a folder (from the Finder): double-click the icon *or* click the icon and select Open from the File menu

To close a folder window (from the Finder): click the close box *or* click the window and select Close from the File menu

To open a folder (from a dialog box): double-click the icon or folder name in the scrolling list

To exit a folder (from a dialog box): click and hold down mouse button on the folder icon in the box; select the level via the pop-up menu that appears

Working with Files

To launch HyperCard or any application: double-click the icon *or* click the icon and select Open from the File menu

To select several files or folders at once: drag a rectangle around the icons *or* click one and Shift-click others

To copy file(s) between disks: drag file icon(s) into the window or icon of the destination disk

To remove file from disk: drag the icon into the Trash

To lock or unlock a file: click the icon, select Get Info from the File menu, and click Locked

Printing

To prepare for printing on a network: select Chooser from the Apple menu, click on a printer icon, click on a printer name, click the close box

To print a card: select Print Card from the File menu

To print a stack: select Print Stack from the File menu

To print a report of text fields in stack: select Print Report from the File menu

To capture a paint screen image (Screen 0): press Command-Shift-3

To print the active window (to a dedicated ImageWriter printer): press Command-Shift-4

To print the screen (to a dedicated ImageWriter): press Caps Lock-Command-Shift-4

APPENDIX D

HYPERCARD SURVIVAL SKILLS

To select . . .

unlocked field text: With the Browse tool drag directly from beginning to end of text.
a rectangular-shaped picture: Drag a diagonal around picture with the Selection tool.
an irregularly shaped picture object: Drag around the object with the Lasso tool.
a button: Select the Button tool and click on the desired button.
a field: Select the Field tool and click on desired field.

Note: If selection method doesn't work, the object to be selected may be in the background. Select Background from the Edit menu and try again.

To delete a selected object or picture segment

Press the Delete (Backspace) key.

To copy or move an object to another card

Select the object (see above).
Choose Copy or Cut (picture, text, button, field) from the Edit menu.
Go to the destination card.
Choose Paste (picture, text, button, field) from the Edit menu.

Drag object to position it where desired.

To move an object from the card layer to the background

Select the object (see above).
Choose Copy or Cut (picture, text, button, field) from the Edit menu.
Select Background from the Edit menu (menu bar becomes striped).
Choose Paste (picture, text, button, field) from the Edit menu.
Drag object to position it where desired.
Select Background again to return to the card layer.

To build a button linked to another card

1. Create the button:
   Choose the Button tool.
   Select New Button from the Objects menu.

2. Customize the button:
   Double-click on the new button (or select Button Info... from the Objects menu).
   Type a name for the button (optional).
   Choose the button style (for invisible button, click transparent with Show Name clicked off).
   Click on Icon... if desired, select an icon, click OK.
   Drag the corner of the button to change its size; drag the center to move it.

3. Link the button:
   Double-click on button again, click on Link to....
   If linking to an existing card, go to that card and click This Card button.
   If linking to another stack, open that stack and click This Stack.
   If linking to a new card with same background, select New Card from the Edit menu and click This Card button.
   If linking to a new card with new background, select New Background from the Objects menu and click This Card button.
   If linking to a new card with a previously used background, go to a card with that background, select New Card from the Edit menu, and click This Card button.

4. Script the button (optional):
   Double-click on the button again, click on Script... button.
   Add to or modify existing script (with special visual effects, sounds, and so on).
   Click OK.

To build a field to contain editable text

1. Create the field:
   Choose the Field tool.
   Select New Field from the Objects menu.
2. Customize the field:
   Double-click on the new field (or select Field Info... from the Objects menu).
   Type a name for the field (optional).
   Choose the field style (transparent for invisible).
   Drag the corner of the field to change its size; drag the center to move it.
   Click on Font..., select desired font and style (use common fonts such as Geneva if your stack will be used on other computers).
3. Fill the field:
   Select Browse (hand) tool (*not* the Paint Text tool), position it over field, and type text.
   Use standard editing techniques, including Cut Text, Copy Text, and Paste Text.

To move a card

Select Cut Card from the Edit menu.
Locate the card in the stack before the position for the new card.
Select Paste Card from the Edit menu.

Note: This will not change the button links for moving between cards unless those links were logical (Go to Next Card); select each button and follow linking procedure under "To build a button."

To add a card or move a card to the beginning of a stack

Paste or create the card so that it follows the current first card in the stack.
Go to the first card in the stack and cut it.
Paste the cut card; it should now be second in the stack, behind the new or moved card.

To save a copy of stack on a different disk

Choose Save a Copy... from the File menu
Click Eject, insert alternate disk, and click Save.
Swap disks as needed.

Note: For maximum protection, do this after each session.

For stylish stacks

Plan your stack design before you build.
Choose a couple of standard fonts to use throughout the stack.
Don't overdo text style effects (outline, shadow, and so on); avoid mixing effects.
Be consistent; use similar buttons, icons, and pictures throughout the stack.
Always provide the user with an exit or Home Button.
Remember that the person reading your stack can't read your mind.

APPENDIX E

ASK DOCTOR HYPER:

A BEGINNER'S HYPERCARD TROUBLESHOOTING GUIDE

Home

HyperCard won't open. What's wrong?

You probably don't have enough available memory. If you're sure that the machine you're using has at least a megabyte of memory, make sure the RAM cache is turned off (see Session 1, page 11).

When I double-click on HyperCard, it responds by asking me, "Where is Home?" Why?

HyperCard needs to have a Home stack to open. It looks for Home in the immediate neighborhood, and if it can't find Home, it asks for help. Locate the Home stack using the dialog box (as explained in Session 1) and double-click on it.

I can't find a button for the ____ stack on my Home card. Where is it?

It's probably nowhere. The Home card is not a complete directory of all your stacks, like the Macintosh Finder. Home comes complete with several buttons for commonly used stacks. If you want a button for another stack on your Home card, you have to install it.

When I click on the _____ button on the Home card, HyperCard asks me, "Where is _____?" Why?

HyperCard looks for stacks in the folders where it has found them before. It keeps this information on one card of the Home stack. If the stack isn't in one of those folders, it will probably ask you to help find it. As long as you don't move that stack later, HyperCard will know next time where to look for it.

Menus

The menu bar is missing. How do I get it back?

To make the menu bar disappear or reappear, press Command-Space.

I only have three (or four) menus. What's wrong?

Your user level is set to one of the first three levels. If you need to create stacks or modify buttons, fields, or scripts, set the user level to scripting.

I see seven menus, but the Objects menu is missing. Where is it?

A paint tool is selected, so the Objects menu has been replaced by the Paint menu, the Options menu, and the Patterns menu. To see the Objects menu, select one of the three tools at the top of the Tools palette.

I can't find the Paint menu, the Options menu, or the Patterns menu. How can I get them back?

Click on any of the paint tools. See previous answer.

Getting Around

When I select Recent from the Go menu, I see the cards in my stack, but they aren't in the right order. How can I rearrange them?

Your cards may, in fact, already be in the right order. The Go Recent command doesn't show you the cards in stack order, but in the order you last visited them. Go Recent is like reviewing a roll of film on which you've taken a snapshot of each new card that you visited.

Typing

When I move the pointer over text that I want to edit, it doesn't turn into an I-beam. Why?

The text is probably paint text, which can't be edited like field text, but must be repainted instead. Another possibility is that the the text is in a locked field; check the Field Info for the field. A third possibility is that a button is sitting on top of the field; hold down the Option and Command keys to see if a button is outlined there.

I know that one of the cards has the phrase "XYZ" written on it; I've seen it. But when I issue a Find "XYZ" command, HyperCard just beeps at me. Why can't it find that phrase?

The Find command will find the card with "XYZ" on it, but only if (1) "XYZ" is on a card in the *current stack*, and (2) "XYZ" is *field text*, not paint text. If HyperCard can't find it, then you can be sure that one of these two conditions hasn't been satisfied.

Painting

When I try to erase my picture, part of it won't go away. Why?

The eraser tool removes only painted material from the current layer. The part that you can't erase is probably either an icon, which can be removed only with the Button tool, or painting on the background, which can be erased only after you select Background (Command-B).

When I try to move an object that I've drawn, I have trouble separating it from its surroundings. Why won't it move easily like objects in Works or MacDraw?

HyperCard's graphics are bit-mapped (like MacPaint), rather than object-oriented (like MacDraw). In plain English, that means that whatever you paint is stored in the computer as just a collection of dots on the screen. When you add a new graphic flourish to your card, the computer initially sees it as a separate flourish, so you can remove it with the Undo command or select and edit it with the Select command. But as soon as you do anything else, the flourish ceases to exist as a separate entity; it becomes just another part of the bit-mapped dot pattern that makes up the graphic

layer of the card. From that point on, it can only be edited as part of the bigger picture.

I created a new card and redesigned its background to show something different from what's on the other cards in the stack. To my dismay, I found out too late that the new background is now on all the old cards. Where did I go wrong?

The New Card command assigns the background of the current card to the new card, so if you change the new card's background, you're also changing all other cards with that background. Instead of selecting New Card, you should have selected New Background. That command creates a new card, too, but it assigns a new background that's completely unrelated to the background of the other cards in the stack.

APPENDIX F

HYPERCARD KEYBOARD COMMANDS

To . . .	*Type*	*Or Select from a Menu*
Go to the last card visited	Command-tilde or down-arrow	Go Back
Go forward through retraced cards	up-arrow	
Go to the previous card in the stack	Command-2 or left-arrow	Go Prev
Go to the next card in this stack	Command-3 or right-arrow	Go Next
Go to the first card in the stack	Command-1 or up-arrow	Go First
Go to the last card in this stack	Command-4 or down-arrow	Go Last
Go to the Help stack	Command-?	Go Help
Go to the Home stack	Command-H	Go Home
Show most recent 42 cards visited	Command-R	Go Recent
Open a new stack	Command-O	Open Stack...
Push a card (store current card's location for quick return)	Command-down-arrow	
Pop a card (return to stored location)	Command-up-arrow	
Show message box	Command-M	Message
Show message box with Find command	Command-F	Find...
Show or hide Tools palette	Option-Tab	
Show or hide the menu bar	Command-Space	
Show outlines of all buttons	Command-Option	
Show all fields (with Field tool selected)	Command-Shift-Option	
Select Browse tool	Command-Tab	Browse tool icon
Select Button tool	Command-Tab-Tab	Button tool icon
Select Field tool	Command-Tab-Tab-Tab	Field tool icon
Undo last action (with many exceptions)	Command-Z	Undo

To . . .	*Type*	*Or Select from a Menu*
Cancel current action (such as printing)	Command-Period	
Copy selected picture, button, or field to Clipboard	Command-C	Copy
Cut selected picture, button, or field to Clipboard	Command-X	Cut
Paste picture, button, field, or card from Clipboard	Command-V	Paste
Paste miniature picture of card from Clipboard	Command-Shift-V	
Paste field, text included	Command-Shift-V	
Show only card picture (with paint tool selected)	Option-D	
Show opaque areas (with paint tool selected)	Option-O	
Select the last graphic object created	Command-S	Select
Select all of the graphics on this layer	Command-A	Select All
Save current picture for next Revert	Command-K	Keep
Revert to picture saved with last Keep	Command-R	Revert
Display the Text Styles dialog box	Command-T	Text Styles...
Bring field or button in front of another object	Command-Plus	Bring Closer
Send field or button behind one object	Command-Minus	Send Farther
Create a new card	Command-N	New Card
Delete the current card	Command-Delete (Backspace)	Delete Card
Print Card	Command-P	Print Card
Quit HyperCard, return to Finder	Command-Q	Quit HyperCard

GLOSSARY

Apple key: See **Command key.**

background: That which is shared by a number of cards; a card's basic template. The background is composed of the **background picture, background fields**, and **background buttons.**

background button: A button that appears on, and whose actions are the same for, all cards with the same background. Contrast with **card button**.

background field: A field whose size, position, and text attributes remain constant on all cards associated with a particular background, but whose text changes from card to card. Contrast with **card field**.

background picture: A picture that applies to a series of cards. You see the Background picture by choosing Background from the Edit menu. Contrast with **card picture**.

browse: To wander through HyperCard's stacks.

Browse tool: The tool you use to click buttons and to set the insertion point in fields.

button: Something that initiates a HyperCard action (making a connection, launching an application, starting a visual or sound effect, and so on) when you click it. See also **background button, card button**.

button action: That which happens when a button is pressed.

Button tool: The tool you use to create, change, and select buttons.

card: HyperCard's basic entity—one screenful of information.

card button: A button that appears on, and whose actions apply to, a single card. Contrast with **background button**.

card field: A field whose size, position, text attributes, and contents are limited to the card on which the field is created. Contrast with **background field**.

card picture: A picture that applies to a specific card. Contrast with **background picture**.

Command key: The key at the lower-left side of the keyboard that has a propeller-shaped symbol. On some keyboards this key also has an Apple symbol and might be called the *Apple key*.

current: (adj.) The one you're using now. For example, the current card is the one you can see on your screen.

field: The place where you type regular (as opposed to Paint) text. Also, the tool you use to create a field. HyperCard has two kinds of fields—**card fields** and **background fields**.

Field tool: The tool you use to create, change, and select fields.

General tool: Any HyperCard tool that isn't a Paint tool. The General tools are Browse, Button, and Field.

Home card: A special card that acts as a pictorial index to stacks. Choose Home from the Go menu to get to Home (or press Command-H).

HyperTalk: HyperCard's built-in script language for HyperCard users.

keyboard equivalent key: A key you press together with the Command key to issue a menu command.

layer: The order of a button or field relative to other buttons or fields on the same card and background. The object created most recently is ordinarily the topmost object (that is on the front layer).

MacPaint document: Document created by the Macintosh MacPaint application, or any application that creates MacPaint-compatible graphics.

message: A command you send to HyperCard through a **script** or through the Message box. Some examples of HyperTalk messages are `Go`, `Show`, `Hide`, `Pop Card`, and `Push Card`.

on-line help: Assistance you can get from an application program while it's running. In this user's guide, on-line help refers to HyperCard's disk-based Help system.

Paint text: Text you type using the Paint Text tool. Paint text can appear anywhere, while **regular text** must appear in a field created with the Field tool. When you finalize Paint text by clicking, it becomes part of a picture.

Paint tool: Any HyperCard tool you use to make pictures. Tools include Lasso, Brush, Spray, Eraser, Paint Text, and many others.

palette: The name for a **tear-off menu** when it's been torn off. A palette remains visible on the screen so you can use it without having to pull down the menu. HyperCard has two palettes—Tools and Patterns.

picture: Any graphic or part of a graphic created with a Paint tool. Also, an imported MacPaint document or part of a MacPaint document.

point: In printing, the measurement of the height of a text character; one point is about 1/72 of an inch. When you select a font, you can also select a point size, such as 10-point, 12-point, and so on.

power key: One of a number of keys on the Macintosh keyboard you can press to initiate a menu action when a Paint tool is active. Power keys are enabled when you choose Power Keys from the Options menu or you check Power Keys in the User Preferences card in the Home stack.

Recent: A special dialog box that holds pictorial representations of the last 42 unique cards viewed. Choose Recent from the Go menu to get the dialog box.

regular text: Text you type inside a field. You use the Browse tool to set an insertion point in a field and then type. Regular text is editable and searchable, while **Paint text** is not.

script: A series of commands written in **HyperTalk** and associated with a particular object.

search path: The route the computer must follow to retrieve a file you ask for.

stack: A group of cards based on the same theme. A HyperCard document. See also **card**.

tear-off menu: A menu that you can convert to a **palette** by dragging the pointer beyond the menu's edge. HyperCard has two tear-off menus—Tools and Patterns.

text field: See **field**.

text property: A quality or attribute of a character's appearance. Properties include style, font, and size.

tool: An implement you use to do work. HyperCard has tools for browsing through cards and stacks, creating text fields, editing text, making buttons, and creating and editing pictures.

user level: The setting on the User Preferences card in the Home stack that lets you use HyperCard's tools and abilities. Five user levels are available: Browsing, Typing, Painting, Authoring, and Scripting.

Index

⌘key. *See* Command key
– (Double dash), 197
“ ” (Quotation marks), 197
¬ (Soft return), 197
~ (Tilde) key, 28, 31, 76, 87

A

about . . . button, 27–28
Absolute links, 118
Absolute reference, defined, 118
Active, defined, 12
Active window, selecting multiple icons in, 41
Activision, 236
Adding
  branches, 172, 174–175
  card layer illusions, 201–202
  cards, 58–59, 245
  emergency exits, 97–98
Addison-Wesley, 233
Address icon, illustrated, 99
Address stack, browsing through, 57–58
Ahouse, Jeremy, 231
Alignment, defined, 81
Amanda Stories, Volume I (Voyager Software), 236
Analyzing structure, 152–153
Animated sequences, scripting, 212–213
Animation, 202–217
Animation aids, 231–232
Animation Help (Ahouse), 231
Answer button, 219–220
Anticipating interactive presentations, 219–220
Ants, marching. *See* Selection rectangle
Apple menu, 9–12
  Calculator, 10
  Chooser, 63
  Control Panel, 10–12
  Scrapbook, 137
Apple Scanner (Apple), 231
Apple's User Group Hotline, 237
Arrow buttons, illustrated, 22, 116
Arrow keys, 30, 31
Art Ideas icon, illustrated, 27, 136, 154
Atkinson, Bill, 230, 236
Authoring (user level), 70–71
Auto hilite (New Button command), 85
Automatic lasso buttons, scripting, 215–217
Avoiding HyperTalk hazards, 222–224

B

Back command (Go menu), 28, 31, 76, 87
Background(s)
  benefits of, 107
  drawing in, 109–114
  of Home stack button/icon, 106–107
  new, 118–121
  old, 138
  rearranging layers in, 199
  what goes in, 152
Background buttons
  borrowing, 116–118
  creating, 196–199
Background command (Edit menu), 106–107
Background layer, buttons on, 155
Background(s), creating, 194–195
Backing up disks, 42–43
`barn door close`, 198
`barn door open`, 198
BASIC, 188
Benefits of backgrounds, 107
Berkeley Macintosh User Group (BMUG), 235, 237
Bird Anatomy, 235
`black`, 199
BMUG (Berkeley Macintosh User Group), 235, 237
Bond, Gary, 238
Books, 237–238
Booting Macintosh, 7–8
Borrowing
  background buttons, 116–118
  buttons, 98–100
  clip art, 136–137
  Find button, 156–157
  Help button, 157–158
  old backgrounds, 138
Boston Computer Society, 235
Branches
  adding, 172, 174–175
  creating, 172–174
  defined, 150
Breaking problems into pieces, 149–150
Broderbund Software, 235
Browse tool, FatBits and, 112, 113
Browsing
  with buttons, 26–28
  through address stack, 57–58
Brush Shape . . . command (Options menu), 90
Bugs, 132
Bullet charts
  creating, 200–201
  dynamic, 195–196
Bulletin boards, 236
Button Ideas icon, illustrated, 156
Button Ideas stack, 156
Button menu (graphic), 169–172
Buttons
  background, 116–118, 196–199
  borrowing, 98–100
  browsing with, 26–28
  card layer vs. background layer, 155
  creating, 83–84, 244
  hidden, 30–31, 94–96, 174
  lasso, 215–217
  linked, 190
  linking, 84–86
  naming, 95
  placeholder, 157
  pop-up field, 192
  radio, 11
  sticky, 174
  testing, 176
  See also specific button
Button tool, 83, 94, 123, 126

C

C, 224
Calculator (Apple menu), 10
Calendar, locating, 47–50
Calendar icon, illustrated, 48
Cancel button, 19
`card`, 199, 220
Card Info . . . command (Objects menu), 190
Card layer, buttons on, 155
Cards
  adding, 58–59, 245
  copying, 134–135
  deleting, 58–59
  drawing, 96
  moving, 245
  title, 194–195
  See also specific card
"Cartoon dialog balloon" icon. *See* about . . . button
Changing
  links, 141–142
  memory of Home card/stack, 46–47
  user level, 45–46
  user name, 45–46
  view in windows, 13–14
Charts (bullet), 195–196, 199–201
`checkerboard`, 198
Checking relative link scripts, 190–191
Chicago 12, 164
Chooser (Apple menu), 63
Claris, 137, 230
Click Art (T-Maker Software), 231
Clicking
  Command key and, 61
  defined, 9
Clip art, 136–137
Clip Art icon, illustrated, 27
Clipboard, 53–57, 86–88
Close box, defined, 12
Close command (File menu), 12
Closing windows, 14
Cloverleaf symbol (⌘) key. *See* Command key
Command-1 (Go First), 30, 31
Command-2 (Go Prev), 30, 31
Command-3 (Go Next), 29–30, 31
Command-4 (Go Last), 30, 31
Command-? (Go Help), 29, 31
Command-A (Select All), 87
Command-B (Background), 106–107
Command-C (Copy), 53, 87
Command-click, 61
Command-Delete (Delete Card), 59
Command-drag, 61, 110
Command-F (Go Find), 60–61
Command-H (Go Home), 28, 31
Command-K (Keep), 76
Command key, 28–29, 87, 110
Command-M (Message), 184
Command-N (New Card), 58, 85–86, 115
Command-N (New Folder), 42
Command-O (Open Stack . . .), 31
Command-Option (show buttons), 30
Command-Q (Quit HyperCard), 31
Command-R (Go Recent), 29, 31
Commands
  Clipboard, 53
  keyboard, 31, 251–252
  See also specific command
Command-S (Select), 87
Command-Space (hide/show menu bar), 30, 31, 120
Command-T (Text Style . . .), 80–81
Command-Tilde (Go Back), 28, 31, 76, 87
Command-V (Paste), 53
Command-X (Cut), 53
Command-Z (Undo), 55, 76
Comment marker, defined, 197
Comments, defined, 197
Compact Stack command (File menu), 222
Compiled languages, defined, 223
CompileIt! (Heizer Software), 234
Compilers, defined, 223
Complete HyperCard Handbook, The (Goodman), 237
CompuServe, 237
Constraint key, 76
Control Panel (Apple menu), 10–12
Copy(ies) (master/working), 36
Copy command (Edit menu), 53
Copying
  cards, 134–135
  files, 39–41
  objects, 243–244
  stacks, 245
Copy Picture command (Edit menu), 87
Cosmic Osmo (Activision), 236
Crissey, Dr. Brian L., 236
Critiquing stacks, 176–177
Curve tool, illustrated, 127
Customizing work space, 44–45
Cut command (Edit menu), 53

D

Database, HyperCard as, 36, 57–64
Data file, defined, 36
Data structure, defined, 220
Debugging, 222–224
Default option, defined, 43
Defining problems, 148
Delete Card command (Edit menu), 59
Delete key, 59
Deleting
  cards, 58–59
  graphics, 135–136
  objects, 243
  text, 52
Designing hypertext, 148–154
Desk accessories. *See* Apple menu *or specific desk accessory*
DeskPaint (Zedcor), 230
Desktop, defined, 8
Desktop Publishing Advisor (Broderbund Software), 235
Dialer (phone), 61–63
Dialog box, defined, 16
Dialoger (theResult Software), 234
Director (MacroMind), 232

Disk(s)
backing up, 42–43
flexible, 5
floppy, 5, 6–7
hard, 5, 6–7, 239
initializing, 37–39
requirements, 239–240
startup, 8
working with, 241
See also specific disk
Diskettes, defined, 5
`dissolve,` 198
`doMenu,` 189
Doors, shaping, 121–123
Double-click-drag, defined, 53
Double-clicking, defined, 12
Double dash (--), 197
Down-arrow key. *See* Arrow keys
Dragging
Command key and, 61, 110
defined, 9
graphics, 110
Option key and, 110
Shift key and, 110
Draw Centered command (Options menu), 141
Draw Filled command (Options menu), 91
Drawing
in background, 109–114
cards, 96
labels, 92–94
lines, 73–74
polygons, 75–77
rooms, 91–92
Draw Multiple command (Options menu), 141
Drive button, 16
Dub'l Click Software, 231
Dynamic bullet charts, creating, 195–196

E

Editing text, 52–53
Edit menu
Background command, 106–107
Copy command, 53
Copy Picture command, 87
Cut command, 53
Delete Card command, 59
New Card command, 58, 85–86, 115
Paste command, 53
Select All command, 41, 87
Select command, 87
Text Style . . . command, 80–81, 114
Undo command, 55, 76
Edit Pattern . . . command (Options menu), 121–122
Effects (visual), 196–199
Eject button, 16
Eject command (File menu), 14
Electronic Arts, 230, 232
Emergency exits, adding, 97–98
Empty introductions, creating, 194
`end mouseUp,` 189
Engineering software, 97, 132–134
Entering text into fields, 50–52
Eraser tool, 74, 112
Esc key, 28, 31, 76, 87
Events Day by Day (Heizer Software), 235
Examining
linked button scripts, 190
scripts of stacks, 186–188
Exit button, 155–156
Exits (emergency), 97–98
External command (XCMD), defined, 223–224
External hard disk, defined, 5

F

Farallon, 232, 233
FatBits command (Options menu), 112–113
Field Info . . . command (Objects menu), 159–160
Fields
creating, 244–245
defined, 36, 50
entering text into, 50–52
fonts in, 164
new, 161–162
pop-up, 162–165, 192
scrolling, 174
text, 50–57
Field (regular) text, 79, 164
Field tool, 158–159
File(s)
copying, 39–41
data, 36
locating in folders, 17–20
working with, 242
See also specific file
File menu
Close command, 12
Compact Stack command, 222
Eject command, 14
Import Paint . . . command, 137
New Folder command, 42
New Stack command, 71
Open command, 12
Open Stack . . . command, 15–17, 31
Page Setup command, 63
Print Card command, 63
Print Report command, 63
Print Stack command, 63
Quit HyperCard command, 31
Save a Copy . . . command, 57
File server
defined, 5
requirements, 239
File server users, tips for, 5–7
Find button
borrowing, 156–157
illustrated, 156
script, 188–189
Find . . . command (Go menu), 60–61
Finder, defined, 8
Finding
hidden buttons, 30–31
mouse, 215
First command (Go menu), 30, 31
Flexible disks, defined, 5
Floppy disks, defined, 5
Floppy-disk users, tips for, 6–7
Flowchart. *See* Map
Flying cards icon, illustrated, 27, 57
Focal Point II (TENpoint0), 235
Folders
locating files in, 17–20
organizing with, 42
working with, 241–242

Font/DA Mover, 164
Fonts, 164
Forget-me-knot reminder icon, illustrated, 55
Format, defined, 37
Freedom Trail, The, 235
Free space, defined, 222
Function, defined, 185. *See also specific function*

G

Geneva font, 164
GEnie, 237
Geography, Maps, illustrated, 203
Getting around, troubleshooting, 248
Go Back key, 28, 31, 76, 87
Go menu, 28
  Back command, 28, 31, 76, 87
  Find . . . command, 60–61
  First command, 30, 31
  Help command, 29, 31
  Home command, 28, 31
  Last command, 30, 31
  Message command, 184
  Next command, 29–30, 31
  Prev command, 30, 31
  Recent command, 29, 31
Goodman, Danny, xi, 235, 237
`go to next card`, 191
Graphic button menu, creating, 169–172
Graphics
  deleting, 135–136
  dragging, 110
  rearranging, 135–136
  selecting, 87, 130
  shrinking, 110
  stretching, 110
Graphic tools, 229–231
`gray`, 199
Grid command (Options menu), 93
Guided Tour, 4

H

"Hand rearranging cards" icon. *See* Sort button
Hard copy, defined, 63
Hard disk(s)
  external, 5
  internal, 5
  requirements, 239
Hard disk users, tips for, 5, 6–7
Hazards
  of HyperTalk, 222–224
  of hypertext, 174
Heading fields, creating, 158–161
Heizer Software, 231, 234, 235, 236
Heizer Software's Stack Exchange, 237
Help button, borrowing, 157–158
Help command (Go menu), 29, 31
Help stack, 23–25
Helvetica font, 164
HFS (Hierarchical File System), 19–20
Hidden buttons, finding, 30–31
`hide`, 187
Hiding menu bar, 30, 31, 120
Hierarchical File System (HFS), 19–20
Hierarchy (object), 189
Highlighting, defined, 9
Home button/icon
  background of, 106–107
  illustrated, 14, 27, 99, 156
Home card/stack
  changing memory of, 46–47
  defined, 15
  illustrated, 15
  troubleshooting, 247–248
Home command (Go menu), 28, 31
Housekeeping, 222–224
HyperCard Developer's Guide (Goodman), 237
HyperCard Script Language Guide: The HyperTalk Language, 238
HyperCard Stack Design Guidelines, 238
HyperCard Visual Animation Tricks (Heizer Software), 231
HyperComposer (Addison-Wesley), 233
HyperDA (Symmetry Corporation), 235
HyPerFormance, 236
Hypermedia, defined, 148
HyperTalk, 184–186, 188, 222–224, 233–234
HyperTalk (Poole), 238
HyperTalk Beginner's Guide: An Introduction to Scripting, 238
Hypertext
  defined, 148
  designing, 148–154
  hazards of, 174
HyperTutor (teligraphics), 235

I, J

I-beam, 80
Icons
  defined, 8
  multiple, 41
  See also specific icon
If Monks Had Macs, 235
Illusions (card layer), 201–202
Import Paint . . . command (File menu), 137
Individual Software, 236
Information services, 236
Inigo Gets Out, 236
Initialize, defined, 37
Initializing disks, 37–39
Inserting text, 52
Insertion bar, defined, 51
Interactive presentations, anticipating, 219–220
Interface. *See* User interface
Internal hard disk, defined, 5
Interpreted language, defined, 223
Interpreter, defined, 223
Intro button/icon, 20–21
Introduction(s), 165–169
  creating, 166–169
  empty, 194
  refining, 151–152
`inverse`, 199
Invert command (Paint menu), 173
Invisible buttons, 94–96, 174
`iris close`, 198
`iris open`, 198

K

K, defined, 38
Keep command (Options menu), 76
Keyboard commands, 31, 251–252. *See also* Command key *or specific command*
Keyboard/keys, navigating by, 28–30. *See also* specific key
Keyboard shortcuts, 28. *See also* Command key *or specific shortcut*

L

Lab administrator's guide, 239–240
Labels, drawing, 92–94
Language(s)
  compiled, 223
  interpreted, 223
  object-oriented, 188
  procedural, 188
Lasso buttons (automatic), 215–217
Last command (Go menu), 30, 31
Launching HyperCard, 14–15
Layer(s)
  background, 155, 199
  card, 155, 201–202
Left-arrow button. *See* Arrow buttons
Left-arrow key. *See* Arrow keys
Legacy of Graphic Design, A (Heizer Software), 236
Level. *See* User level
LightningScan (Thunderware), 231
Lines, drawing, 73–74
Line Size . . . command (Options menu), 88
Linked button scripts, examining, 190
Linking buttons, 84–86
Links
  absolute, 118
  changing, 141–142
  defined, 85
  logical, 118, 190–191
  two-way, 123–127
Locating
  Calendar, 47–50
  files in folders, 17–20
Locked, defined, 38
Lock Text (Field Info . . . command), 159–160
Logical links, 118, 190–191
Logical reference, defined, 118
LogiTech, 231
Loose ends, keeping track of, 175–176

M

Macintosh
  booting, 7–8
  preparing, 239–240
  requirements, 239–240
  survival skills, 241–242
Mac the Knife (Miles Computing), 231
MacPaint (Claris), 137, 230
MacRecorder (Farallon), 233
MacroMind, 232
Maller, Steve, 233
Manhole, The (Activision), 236
Map
  defined, 97
  tips, 133
Marching ants. *See* Selection rectangle
Marquee. *See* Selection rectangle
Master copy, defined, 36
Master menu, creating, 154–158
Memory (of Home card/stack), changing, 46–47
Menu(s)
  graphic button, 169–172
  master, 154–158
  pop-up, 19
  pull-down, 9
  tear off, 73
  troubleshooting, 248
  See also specific menu
Menu bar
  hiding/showing, 30, 31, 120
  stripes in, 109
Message box, defined, 60, 184
Message command (Go menu), 184
Message handlers, defined, 188
Messages, 184–186, 188
Miles Computing, 231
Monaco font, 164
Monster Hunt, 236
Mouse
  finding, 215
  using, 8–9
`mouseLoc`, 215
`mouseUp`, 189
Moving
  cards, 245
  furniture, 130–132
  objects, 243–244
Multiple icons, selecting, 41
Murder on Main Street (Heizer Software), 236
Musical cards, creating, 217–219

N

Name (user), 45–46
Naming
  buttons, 95
  objects, 95
  stacks, 72
Navigating by keyboard, 28–30
Networks
  defined, 8, 134
  planning, 132–134
New Background command (Objects menu), 118–119
New backgrounds, creating, 118–121
New button (New Stack command), 72
New Button command (Objects menu), 83, 94, 123–124
New Card command (Edit menu), 58, 85–86, 115
New Field command (Objects menu), 161
New fields, creating, 161–162
New Folder command (File menu), 42
New Stack command (File menu), 71
New stacks, creating, 71–72
New York font, 164
`next card`, 191
Next command (Go menu), 29–30, 31
No paint, 93

O

Object hierarchy (HyperCard), 189

Object-oriented language, defined, 188
Objects, 188
  copying, 243–244
  deleting, 243
  moving, 243–244
  naming, 95
Objects menu, 83
  Card Info . . . command, 190
  Field Info . . . command, 159–160
  New Background command, 118–119
  New Button command, 83, 94, 123–124
  New Field command, 161
  Send Farther command, 202
  Stack Info . . . command, 186
Old backgrounds, borrowing, 138
101 Scripts and Buttons for HyperCard (Individual Software), 236
`on mouseUp`, 189
`on openStack`, 187–188
`openCard`, 218
Open command (File menu), 12
Opening
  stacks, 15–17
  windows, 12
`openStack`, 187–188
Open Stack . . . command (File menu), 15–17, 31
Operating system, defined, 7
Option-drag, 110
Option key, 31, 110, 112, 113, 197
Option-Return, 197
Options menu
  Brush Shape . . . command, 90
  Draw Centered command, 141
  Draw Filled command, 91
  Draw Multiple command, 141
  Edit Pattern . . . command, 121–122
  FatBits command, 112–113
  Grid command, 93
  Keep command, 76
  Line Size . . . command, 88
  Revert command, 76
Organizing with folders, 42
Outlining
  problems, 192–193
  stacks, 152

P

Page Setup command (File menu), 63
Paintbrush tool, illustrated, 90
Paint Bucket tool, illustrated, 78
Painting, troubleshooting, 249–250
Painting (user level), 70
Paint menu, Invert command, 173
Paint patterns, pouring, 77–79
Paint programs, 229–230
Paint text, 79–82, 164
Paint Text tool, 79–82, 114
Paint tool animation, defined, 214
Palettes. *See specific palette*
Pascal, 188, 224
Paste command (Edit menu), 53
Patterns (paint), 77–79
Patterns menu/palette, 77–79
Pencil tool, 73–74, 112, 113
Phone dialer, 61–63
Phone icon, illustrated, 62, 157
Placeholder buttons, creating, 157
Planning networks, 132–134
`play`, 218–219
Pointing hand tool. *See* Browse tool
Points, defined, 81
Polygons, drawing, 75–77
Polygon tool, 75–77, 123
Poole, Lon, 230
`pop card`, 220
Pop-up field buttons, scripts for, 192
Pop-up fields, creating, 162–165
Pop-up menu, defined, 19
Portable furniture, creating, 127–130
Port Authority (Heizer Software), 234
Pouring paint patterns, 77–79
Power Keys, 71, inside back cover
Preparing Macintosh, 239–240
Presentations (interactive), 219–220
Prev command (Go menu), 30, 31
Print Card command (File menu), 63
Printers/printing, 63, 242
Print Report command (File menu), 63
Print Stack command (File menu), 63
Problems
  breaking into pieces, 149–150
  defining, 148
  outlining, 192–193
  See also Troubleshooting
Procedural language, defined, 188
Program(s), 188. *See also* specific program
Property, defined, 192
Protecting HyperCard, 36–37
Pull-down menu, defined, 9
`push card`, 220

Q

Question Mark button/icon, 62, 157
`quickly`, 199
Quick tour, 20–23
Quit HyperCard command (File menu), 31
Quitting HyperCard, 31
Quotation marks (" "), 197
Quotations icon, 26–27

R

Radio button, defined, 11
RAM Cache (Control Panel), 10–11
Rearranging
  graphics, 135–136
  layers in background, 199
Recent command (Go menu), 29, 31
Records, defined, 36
Rectangle tool, illustrated, 91, 119

Reference (absolute/logical), 118
Regular Polygon tool, illustrated, 141
Remembering users, 134
`repeat`, 213
Replacing text, 52
Report, defined, 63
Requirements, 239–240
ResCopy (Maller), 235
ResEdit, 232–233
Resetting user level, 25–26, 70–71
Resources
  defined, 218
  HyperCard, 229–237
  sound, 234–235
Restart command (Special menu), 31
Return (soft), 197
Return-arrow button, 24, 27, 219–220, 221
Return character, defined, 56
Return key, 197
Revert command (Options menu), 76
Right-arrow button. *See* Arrow buttons
Right-arrow key. *See* Arrow keys
Rooms, drawing, 91–92
Roots, defined, 150
Rounded Rectangle tool, illustrated, 92

S

Sample stacks, 236–237
Save a Copy . . . command (File menu), 57
Saving stacks, 245
ScanMan (LogiTech), 231
Scanners, 137, 233
Science, Astronomy, Chem . . . icon, illustrated, 136
Scrapbook (Apple menu), 137
ScréenRecorder (Farallon), 232
ScriptEdit (Somak Software), 234
Script(s)/scripting
  animated sequences, 212–213
  automatic lasso buttons, 215–217
  defined, 186
  examining, 186–188
  linked button, 190
  relative link, 190–191
  shortcuts to, 201
  syntax of, 197
Scroll bars, 13
Scroll box, 13
`scroll down`, 198
Scrolling fields, defined, 174
`scroll left`, 198
`scroll right`, 198
`scroll up`, 198
Select All command (Edit menu), 41, 87
Select command (Edit menu), 87
Selecting, 243
  defined, 9
  graphics, 87, 129
  multiple icons, 41
  text, 52, 53
Selection rectangle
  defined, 87
  illustrated, 88, 110
Send Farther command (Objects menu), 202
Sending HyperTalk messages, 184–186
Sequences (animated), 212–213
Shaping doors/windows, 121–123
Shift-click, defined, 40
Shift-drag, 110
Shift key, 76, 110
Shift-Tab, defined, 58
Shortcuts
  keyboard, 28
  to scripts, 201
  See also Command key *or specific shortcut*
`show`, 187
Showing menu bar, 30, 31, 120
Show name (New Button command), 95
Shrinking graphics, 110
Shut Down command (Special menu), 31
Shutting down, 31
Size, defined, 81
Size box, illustrated, 13
`slowly`, 199
Soft return (¬), 197
Software engineering, 97, 132–134
Somak Software, 234
Sort button, illustrated, 59
Sorting stacks, 59–60
Sound resources, 232–233
Sources (HyperCard), 229–237
Space (free), 222
Space bar, 30, 31, 120
Speaker Volume (Control Panel), 11
Special menu
  Restart command, 31
  Shut Down command, 31
Stack(s)
  copying, 245
  critiquing, 176–177
  defined, 220
  examining scripts of, 186–188
  keeping orderly, 143
  naming, 72
  new, 71–72
  opening, 15–17
  outlining, 152
  sample, 236–237
  saving, 245
  sorting, 59–60
  starting, 108–109
  stylish, 245–246
  testing, 176
  working, 154–155
  See also specific stack
Stack Exchange, 237
Stack Info . . . command (Objects menu), 186
Standard fonts, 164
Star Myths (Crissey), 236
Starting, 37
  at Home card/stack, 45
  HyperCard, 14–15
  stacks, 108–109
Startup disk, defined, 7
State-Smart (HyPerFormance), 236
Stepwise refinement, defined, 150
Sticky buttons, 174
Straight Line tool, 74, 111
Stretching graphics, 110
Stripes (in menu bar), 109
Structure
  analyzing, 152–153
  data, 220
  tree, 150–151, 169–175
Studio/1 (Electronic Arts), 230, 232
Styles, defined, 81
Stylish stacks, creating, 245–246

Survival skills
HyperCard, 243–246
Macintosh, 241–242
Symmetry Corporation, 235
Syntax (of scripts), 197
System files, 7, 164
System Folder icon, illustrated, 12

T

Tab key, 58
Tearing off Tools menu, 72–73
Tear off menu, defined, 73
teligraphics, 235
TENpoint0, 235
Testing
buttons, 176
stacks, 176
Text
deleting, 52
editing, 52–53
entering into fields, 50–52
field, 79, 164
inserting, 52
paint, 79–82, 164
replacing, 52
selecting, 52, 53
Text fields, 50–57
Text Style . . . command (Edit menu), 80–81, 114
theResult Software, 234
`the time,` 185
ThunderScan (Thunderware), 231
Thunderware, 231
Tilde (~) key, 28, 31, 76, 87
`time,` 185
Times font, 164
Tips
FatBits, 113
for file server users, 5–7
Find . . . command, 61
for floppy-disk users, 6–7
for hard disk users, 5, 6–7
map, 133
Title bar, defined, 12
Title cards, creating, 194–195
Titles
creating, 114–116
typing, 158
T-Maker Software, 231
To Do list, 176
Tools menu/palette, tearing off, 72–73. *See also* specific tool
Tour (quick), 20–23
Transparent (New Button command), 95
Trash, 43–44
Tree structure, 150–151, 169–175
Trendware, 234
Tricks. *See* Shortcuts
Troubleshooting, 247–250. *See also* Problems
Two-way links, creating, 123–127
Typing
titles, 158
troubleshooting, 249
Typing (user level), 46

U

Uncovering bugs, 132
Undo command (Edit menu), 55, 76
Unfinished Business list, 176
Up-arrow key. *See* Arrow keys
User Group Hotline, 237
User groups, 236
User interface
defined, 97
designing, 153–154
User level
changing, 45–46
defined, 25–26
resetting, 25–26, 70–71
See also specific user level
User name, changing, 45–46
Users
file server, 5–7
floppy-disk, 6–7
hard disk, 5, 6–7
remembering, 134

V

`venetian blinds,` 198
`very quickly,` 199
`very slowly,` 199
View (in windows), changing, 13–14
View menu, 13
`visible,` 192
"Vision of HyperCard, The" (Sez or Atkinson), 236
`visual effect,` 191, 198
Visual effects, 196–199
Voyager Software, 237

W

Wet Paint (Dub'l Click Software), 231
"What is HyperCard?" button, illustrated, 21–22
"Where is . . . ?", 20
`white,` 199
White paint, illustrated, 93
Window(s)
active, 41
changing view in, 13–14
closing, 14
opening, 12
shaping, 121–123
`wipe down,` 198
`wipe left,` 191, 198
`wipe right,` 198
`wipe up,` 198
Working
with disks, 241
with files, 242
with folders, 241–242
Working copies, defined, 36
Working stack, creating, 154–155
Work space, customizing, 44–45
Write-protected, defined, 38

X

XCAL (Trendware), 234
XCMD (external command), defined, 223–224
XCMD's for HyperCard (Bond), 238

Z

Zedcor, 230
Zoom box
defined, 14
illustrated, 14
zoom close, 198
zoom in, 198
zoom open, 198
zoom out, 198

Power Keys ↓

Paint

Visible only when a paint tool is selected

S	Select ⌘S	Selects the most recently drawn shape or figure
A	Select All ⌘A	Selects the entire card picture (or background picture)
F	Fill	Fills the selection (or last shape drawn) with the current pattern
I	Invert	Inverts the selected picture (black to white, white to black)
P	Pickup	Copies underlying image in the currently selected shape
D	Darken	Makes selected area of picture slightly darker
L	Lighten	Makes selected area of picture slightly lighter
E	Trace Edges	Outlines the black in the selected picture (or in the entire picture)
[	Rotate Left	Rotates the selected picture or the last drawn object 90 degrees left
]	Rotate Right	Rotates the selected picture or the last drawn object 90 degrees right
V	Flip Vertical	Flips the selection or the last drawn object vertically
H	Flip Horizontal	Flips the selection or the last drawn object horizontally
O	Opaque	Makes the selected area of the card picture opaque
T	Transparent	Makes the selected area of the card picture transparent
	Keep ⌘K	Saves the current picture as the one to revert to
R	Revert	Reverts to the last kept version (undoes everything done since then)

Tools and Keyboard Modifiers

Tools	Click on tool	Double-click on tool	Drag with tool
Browse tool (turns to I-beam over unlocked fields	selects Browse tool; used to click buttons, edit field text		selects field text
Button tool	selects Button tool; used to create, modify buttons		from middle moves button; from edge stretches button
Field tool	selects Field tool; used to create, modify fields		from middle moves field; from edge stretches field
Selection tool (marquee)	reveals paint menus	selects entire graphics layer	selects rectangular picture; moves or stretches picture
Lasso	reveals paint menus	selects all images in graphics layer	selects irregular picture (shrink to fit)
Pencil	reveals paint menus	zooms to FatBits at position of last work	draws fine line
Paintbrush	reveals paint menus	displays Brush Shapes dialog box	draws with selected pattern, brush shape
Eraser	reveals paint menus	erases entire graphics layer	erases graphics
Straight Line	reveals paint menus	displays Line Size dialog box	draws straight line
Spray Can	reveals paint menus		paints spray with selected pattern
Rectangle	reveals paint menus	toggles Draw Filled on/off	draws rectangle with selected line width
Rounded Rectangle	reveals paint menus	toggles Draw Filled on/off	draws rounded rectangle with selected line width
Oval	reveals paint menus	toggles Draw Filled on/off	draws oval with selected line width
Curve	reveals paint menus	displays Polygon Sides dialog box	draws irregular curved shapes with selected line width
Regular Polygon	reveals paint menus	toggles Draw Filled on/off	draws regular polygons with selected line width
Polygon	reveals paint menus	toggles Draw Filled on/off	draws irregular polygons with selected line width
Paint Bucket	when clicked in card, fills surrounded graphic area with pattern	displays Patterns pallette	
Paint Text tool	reveals paint menus	reveals Text Styles dialog box	click to position pointer for typing paint text

Learn HyperCard in just a few hours.

You can do it easily with the entertaining tutorials in this book. Even the least technical reader can quickly learn to design, build, and customize visually impressive and conceptually complex stacks by following George Beekman's step-by-step directions.

HyperCard in a Hurry is built around six interactive, hands-on sessions. The how-to-do-it instructions are complemented by discussions of design issues that are crucial to success in developing software.

Each session builds on concepts presented in the previous session and is followed by a summary of important new concepts, a set of self-testing exercises, and suggested projects.

Also included —

- A beginner's "TroubleShooting Guide"
- Summary of menu commands
- A set of how-to-do-it procedures
- An annotated list of resources for further exploration of HyperCard

ISBN 0-534-13038-0

90000

9 780534 130381